

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Janez Alič

Krmilni sistem za majhne piščančje farme

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Janez Alič

Krmilni sistem za majhne piščančje farme

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Patricio Bulić

Ljubljana, 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Načrtujte krmilni sistem za majhne piščančje farme. Krmilni sistem naj bo zgrajen okrog računalnika Raspberry Pi. Sistem naj uporablja senzorje temperature in nivoja vode ter naj ju krmili po programu, ki je priporočen za določene starosti piščancev. Izdelajte tudi spletno aplikacijo za nadzor in krmiljenje sistema.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Janez Alič, z vpisno številko **63090063**, sem avtor diplomskega dela z naslovom:

Krmilni sistem za majhne piščančje farme

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Patricia Bulića,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 7. oktobra 2014

Podpis avtorja:

Zahvaljujem se mentorju izr. prof. dr. Patriciu Buliću za pomoč in potrpežljivost pri izdelavi diplomskega dela. Posebna zahvala gre tudi moji družini in ostalimi, ki so mi stali ob strani tekom študija.

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	1
Poglavje 2	Predstavitev predhodnega stanja	3
2.1	Krmiljenje vode	3
2.2	Krmiljenje svetlobe	4
2.3	Krmiljenje temperature	4
2.4	Opis problema	5
Poglavje 3	Računalnik Raspberry Pi.....	7
3.1	Predstavitev uporabljenih tehnologij	7
3.1.1	Raspberry Pi model B revizije 2.....	7
3.1.2	Operacijski sistem Raspbian.....	8
3.1.3	Oddaljen dostop	9
3.1.4	Upravljanje s pini GPIO	11
3.1.5	Programiranje v programskem jeziku Python	13
3.2	Predstavitev izdelane rešitve	13
3.2.1	Uporabljene knjižnice.....	13
3.2.2	Krmiljenje temperature.....	13
3.2.3	Povezava s spletnim vmesnikom.....	14
3.2.4	Upravljanje stanj pinov GPIO	15
3.2.5	Glavna zanka programa	15
Poglavje 4	Elektronsko vezje naprav	19
4.1	Predstavitev uporabljenih tehnologij, orodij in naprav	19
4.1.1	Ohišje.....	19

4.1.2	Napajanje	19
4.1.3	Relejski modul Keyes_SRly	20
4.1.4	Senzor temperature DS18B20.....	20
4.1.5	Senzor nivoja vode.....	21
4.1.6	Elektromagnetni ventil TFW-1S	22
4.2	Predstavitev izdelane rešitve	22
4.2.1	Shema elektronskega vezja	23
Poglavje 5	Spletni vmesnik	25
5.1	Predstavitev uporabljenih tehnologij.....	25
5.1.1	MySQL.....	25
5.1.2	HTML	25
5.1.3	CSS.....	25
5.1.4	PHP	26
5.1.5	Javascript.....	26
5.1.6	jQuery.....	26
5.1.7	JSON	26
5.1.8	Ajax.....	26
5.1.9	Bootstrap	27
5.2	Predstavitev izdelane rešitve	27
5.2.1	Podatkovna baza	27
5.2.2	Vmesnik za upravljanje s podatkovno bazo.....	32
5.2.3	Nadzorna plošča.....	34
Poglavje 6	Sklepne ugotovitve	41

Seznam uporabljenih kratic

kratica	angleško	slovensko
RPI	Raspberry Pi	Raspberry Pi
GPIO	general purpose input output	splošno namenski vhodi in izhodi
SSH	secure shell	varnostna lupina
SoC	system on chip	sistem na čipu
USB	universal serial bus	univerzalno serijsko vodilo
RCA	radio corporation of America	Ameriška radio korporacija
HDMI	high definition multimedia interface	multimedijski vmesnik visoke ločljivosti
DSI	display serial interface	serijski vmesnik za prikaz
I2S	integrated interchip sound	integrirani medčip zvoka
IP	internet protocol	internetni protokol
URL	uniform resource locator	enolični krajevnik vira
XML	extensible markup language	razširljiv označevalni jezik
CSV	comma separated value	vrednosti ločene z vejico
PHP	hypertext preprocessor	predprocesor nadbesedila
SQL	structured query language	strukturirani povpraševalni jezik
HTML	hypertext markup language	jezik za označevanje nadbesedila
CSS	cascading style sheet	kaskadne oblikovne predloge
JSON	javascript object notation	javascript objektni zapis
AJAX	asynchronous javascript and XML	asinhroni javascript in XML
NO	normally open	običajno odprt
NC	normally closed	običajno zaprt

Povzetek

Diplomsko delo opisuje razvoj sistema za samodejno in ročno krmiljenje piščančje farme. Združuje uporabo računalnika RPI (angl. Raspberry Pi), zgrajenega elektronskega vezja ter spletnega vmesnika.

Elektronsko vezje naprav je namenjeno odčitavanju trenutnega stanja nivoja vode v zbiralniku, trenutne temperature in osvetlitve prostora. Vsebuje več vrst senzorjev, relejske module, elektromagnetni ventil ter nekaj osnovnih elektronskih komponent. Povezava vezja z računalnikom RPI je vzpostavljena preko pinov GPIO (angl. general purpose input output). Trenutna stanja naprav se beležijo v podatkovno bazo preko zahtev vmesnika API (angl. application programming interface). Na računalniku RPI se izvaja program v jeziku Python, ki krmili z napravami. Sistem omogoča samodejno krmiljenje naprav na podlagi vnaprej predpisanih pravil, ki se spreminjajo s starostjo piščancev. Prav tako je možno tudi ročno upravljanje znotraj nadzorne plošče sistema. Dostop je omogočen zgolj prijavljenim uporabnikom. Služi za pregled nad trenutnimi in preteklimi stanji. V kolikor se pojavi težava z vzpostavitvijo internetne povezave, se sistem krmili s pomočjo pravil znotraj tekstovnih datotek.

Ključne besede: krmiljenje naprav, Raspberry Pi, elektronsko vezje, spletni vmesnik, podatkovna baza, nadzorna plošča

Abstract

This thesis describes the development of a system for automatic and manual control of chicken farm equipment. The system uses a RPI (Raspberry Pi) computer, with a built in electronic circuit control and a web interface.

The purpose of electronic wiring of components is for reading the water level, temperature and lighting control. This system contains several types of sensors, relay modules, some basic electronic components and a solenoid water valve. These components are connected to the RPI computer via GPIO (general purpose input output) pins. System status is constantly being logged into the database via the API (application programming interface). The RPI computer runs a script in Python programming language, which controls the components. The system provides automatic control based on pre-specified rules, which vary with the age of chickens. Manual control is also possible by changing the settings in the control panel, available to registered users. Via the control panel you also have access to past system statistics data. If internet connection is lost, the RPI computer can be controlled by means of updating data in text files.

Keywords: controlling devices, Raspberry Pi, electronic wiring, web interface, database, control panel

Poglavje 1 Uvod

Razvoj in potreba po uporabi novih tehnologij je dandanes v velikem porastu. Za skoraj vsako opravilo je mogoče z uporabo današnjih tehnologij zgraditi sistem, ki se upravlja samodejno. K temu načinu upravljanja stremi vedno večji odstotek industrije, zato lahko pričakujemo le še vedno več zanimanja za računalniško vodene sisteme. Dokaj nedotaknjeno ostaja področje kmetijstva, kjer predvsem lastniki manjših kmetij še vedno veliko dela opravijo sami, čeprav gre za povsem rutinska dela, ki bi jih bilo mogoče avtomatizirati. Razlogov za to je več, eden glavnih pa je zagotovo finančna obremenitev vzpostavitve takega sistema.

Za potrebe vodenja manjše piščančje farme smo zgradili sistem, ki se upravlja brez stalnega nadzora skrbnika. S tem smo močno razbremenili skrbnika in povečali uspešnost reje. Poleg omenjenih prednosti, nam zgrajen sistem omogoča samodejno upravljanje naprav na podlagi vnaprej predpisanih pravil, ter ročno upravljanje s sistemom preko spletnega vmesnika 24 ur na dan. Ne glede na to kje smo, lahko pregledujemo trenutna stanja ter upravljamo z napravami.

V diplomskem delu smo v začetnem poglavju opisali predhoden način vodenja piščančje farme. Predstavljene so naprave, ki še vedno služijo svojemu namenu tudi z vpeljavo našega sistema. Spremenilo se je zgolj upravljanje z napravami. Sledi opis našega zgrajenega sistema skozi predstavitev uporabljenega računalnika RPI, elektronskega vezja naprav in spletnega vmesnika. Predstavljene so vse uporabljene tehnologije. V zaključnem delu pa smo predstavili še naše rezultate ter povzeli nekaj sklepnih misli o našem projektu.

Poglavje 2 Predstavitev predhodnega stanja

Predhoden način vodenja piščančje farme je od skrbnika zahteval, da je večkrat na dan obiskal prostor v katerem je vzrejal piščance ter preveril trenutne bivalne pogoje in stanja naprav. Na podlagi odčitanih pogojev in vnaprej predpisanih pravil pa je ustrezno ukrepal. Največji problem je bil stalno zagotavljanje idealnih pogojev.

2.1 Krmiljenje vode

Na površini celotnega prostora je porazdeljenih več napajalnikov za vodo, ki se napajajo preko večjega zbiralnika za vodo (Slika 2.1). Zelo pomembno je, da je v zbiralniku vedno zadosti vode, zato mora skrbnik stalno nadzorovati nivo vode. V kolikor nivo vode preveč pade, mora zbiralnik napolniti, saj v nasprotnem primeru ostanejo napajalniki brez oskrbe z vodo.



Slika 2.1 Prikaz uporabljenega sistema za krmiljenje vode. S številko 1 je označen zbiralnik, s številko 2 pa napajalnik za vodo.

2.2 Krmiljenje svetlobe

Za osvetlitev prostora skrbi več žarnic, ki so enakomerno porazdeljene po celotnem prostoru. Žarnice so krmiljene z uporabo nastavljivega časovnika (Slika 2.2). Krmiljenje svetlobe določa pravilo, ki se spreminja s starostjo piščancev in trenutnim časom.



Slika 2.2 Prikaz uporabljenega časovnika za krmiljenje svetlobe.

Za piščanca, starega od 0 do 7 dni je potrebno zagotoviti prostor, ki je osvetljen vseh 24 ur na dan, za starejšega, pa je priporočeno, da je prostor 1 uro na dan neosvetljen oziroma v temi [1], [2]. Ker je osvetlitev prostora odvisna tudi od zunanjih vremenskih razmer in letnih časov, je potrebno stalno spreminjanje nastavitve časovnika.

2.3 Krmiljenje temperature

Za preverjanje trenutne temperature v prostoru se uporablja termometer, ki je postavljen na višini piščancev 20 cm od tal. Na podlagi odčitane temperature in pravil, ki se spreminjajo s starostjo piščancev (Tabela 2.1), se skrbnik odloča, ali bo prostor dodatno ogreval ali ohlajeval. Za ogrevanje prostora skrbi električni grelec, za hlajenje pa električni ventilator.

Starost (dni)	0 – 7	8 – 14	15 – 21	22 – 28	29 – 35	36 – 42	32 – 49
Temperatura (°C)	31 – 32	29 – 26	26 – 24	24 – 22	22 – 21	21	21

Tabela 2.1: Priporočena temperatura v odvisnosti od starosti piščancev [1], [2].

2.4 Opis problema

Zagotavljanje idealnih pogojev je ključnega pomena in le tako lahko pričakujemo uspešno rejo. Ker pa vzdrževanje takih pogojev zahteva zelo veliko pozornosti in stalnega nadzora skrbnika 24 ur na dan, je to pravzaprav nemogoče doseči. Potreba po sistemu, ki se krmili samodejno, omogoča pa tudi ročno krmiljenje naprav, je zares velika. Tak sistem bi močno razbremenil skrbnika, prav tako pa bi omogočal tudi boljši pregled nad trenutnim stanjem.

Poglavje 3 Računalnik Raspberry Pi

Iskali smo napravo, ki bo primerna za upravljanje celotnega sistema. Zaradi visoke zmogljivosti, možnosti povezovanja drugih naprav in cenovno ugodne rešitve [3], smo se odločili za uporabo računalnika RPI. Nanj smo namestili orodja za lažje upravljanje z napravami in razvili programsko rešitev sistema.

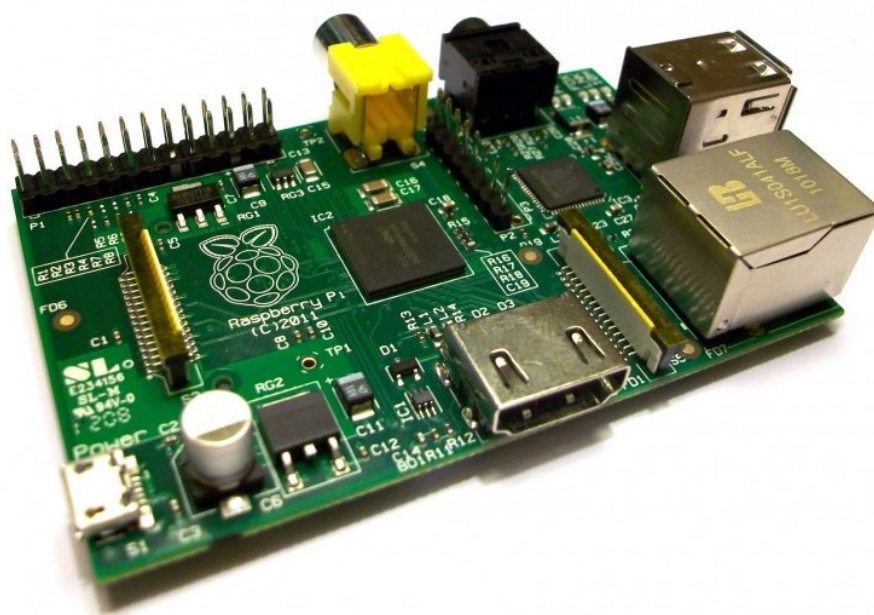
3.1 Predstavitev uporabljenih tehnologij

Ugotovili smo, da je računalnik RPI odlična izbira za krmiljenje našega sistema. Zaradi odprtokodnega sistema je uporabnikom na voljo ogromno knjižnic in drugih orodij, ki poenostavijo proces krmiljenja.

Z namestitvijo operacijskega sistema smo si zagotovili delovno okolje, ki nam je omogočalo nadaljnji razvoj. Ker smo si želeli delo poenostaviti, smo preko terminala namestili še nekaj drugih orodij, ki niso vključeni v osnovnem paketu operacijskega sistema. Uporabili smo program LXTerminal, ki je že nameščen na operacijskem sistemu Raspbian.

3.1.1 *Raspberry Pi model B revizije 2*

RPI je računalnik, velikosti kreditne kartice in teže 45 g, ki je bil izdelan z namenom povečanja zanimanja za računalništvo v šolah [4]. Ker gre za cenovno zelo ugoden ter izredno zmogljiv računalnik [3], je postal zelo priljubljen med navdušenci nad računalništvom. Na trgu je na voljo več različic računalnika RPI, sami pa smo se odločili za RPI modela B revizije 2 [5] (Slika 3.1).



Slika 3.1: Prikaz računalnika RPI [6].

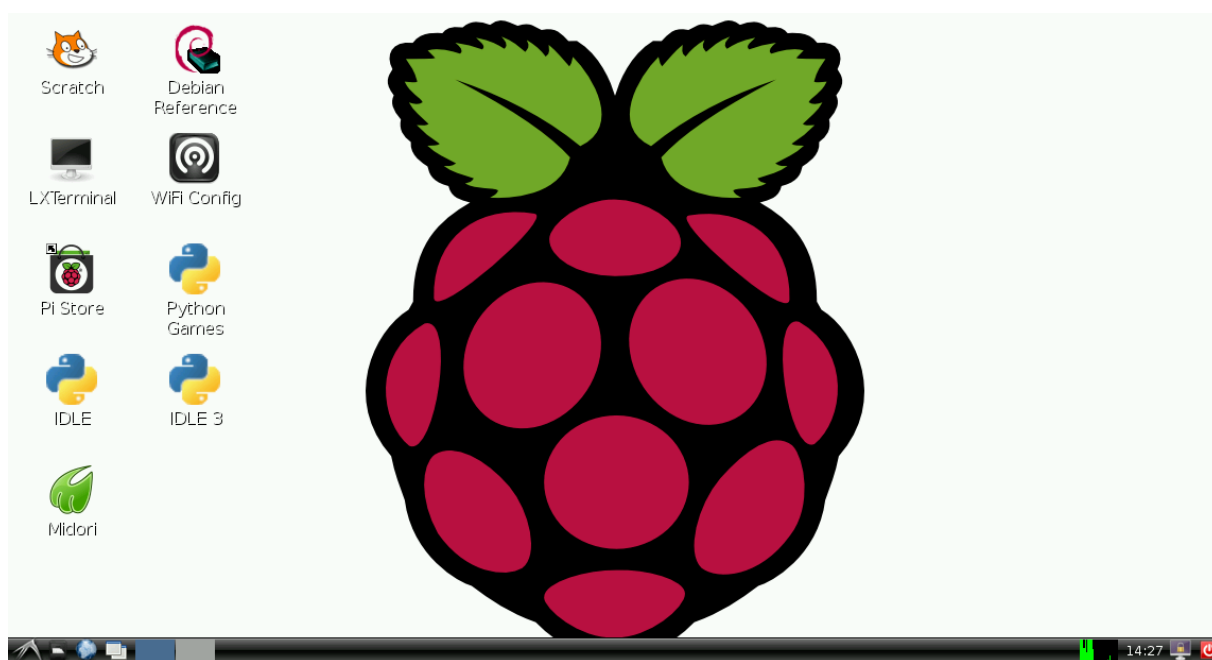
Srce računalnika RPI je Broadcom BCM2835 [7] SoC (angl. system on chip), ki vsebuje procesor ARM1176JZF-S [8] 700 MHz, grafično procesno enoto ter 512 MB delovnega pomnilnika [9]. Za napajanje potrebuje zgolj 5 V / 1 A, ki jo prejema preko priključka mikro USB (angl. universal serial bus). Ker ne vsebuje vgrajenega trdega diska, potrebuje za delovanje in shranjevanje podatkov spominsko kartico SD. Povezava s spletom je mogoča preko vhoda Ethernet RJ45. Ima kar tri različne načine za priklop izhodnih video naprav, in sicer preko kompozitnega RCA (angl. radio corporation of America), HDMI (angl. high definition multimedia interface) ter DSI vhoda (angl. display serial interface). Za izhod zvoka prav tako skrbi več izhodov, in sicer HDMI, 3,5 mm jack ter I²S (angl. integrated interchip sound). Upravljanje naprav v večini poteka preko vhodov oz. izhodov GPIO. Poleg omenjenih vhodov in izhodov, pa ima računalnik RPI še dva vhoda USB, v katera lahko priključimo miško in tipkovnico ali pa druge vhodno-izhodne naprave.

3.1.2 Operacijski sistem Raspbian

Tudi računalnik RPI za komunikacijo med uporabnikom in strojno opremo uporablja operacijski sistem. Na voljo je veliko različnih operacijskih sistemov, ki jih je mogoče namestiti. Najbolj pogosto uporabljen je tako imenovan Raspbian, ki temelji na operacijskem sistemu Linux distribucije Debian [4]. Ker gre za operacijski sistem prilagojen uporabi računalnika RPI, smo se tudi mi odločili za uporabo slednjega.

Sliko operacijskega sistema smo prenesli iz uradne strani projekta RPI [10], kjer se nahaja še nekaj drugih podprtih operacijskih sistemov. Na spominsko kartico SD smo jo namestili z uporabo programa Win32DiskImager. Programa je brezplačen in ga je mogoče prenesti iz uradne spletne strani [11] projekta Win32DiskImager.

V režo za spominsko kartico na računalniku RPI smo umestili našo kartico z nameščenim operacijskim sistemom. Preko USB vhodov smo povezali še miško in tipkovnico, preko HDMI vhoda pa monitor. Ko smo povezali še napajalni kabel z virom napetosti, se je na monitorju pojavil uporabniški vmesnik operacijskega sistema, ki je zelo podoben preostalim Linux operacijskim sistemom (Slika 3.2).

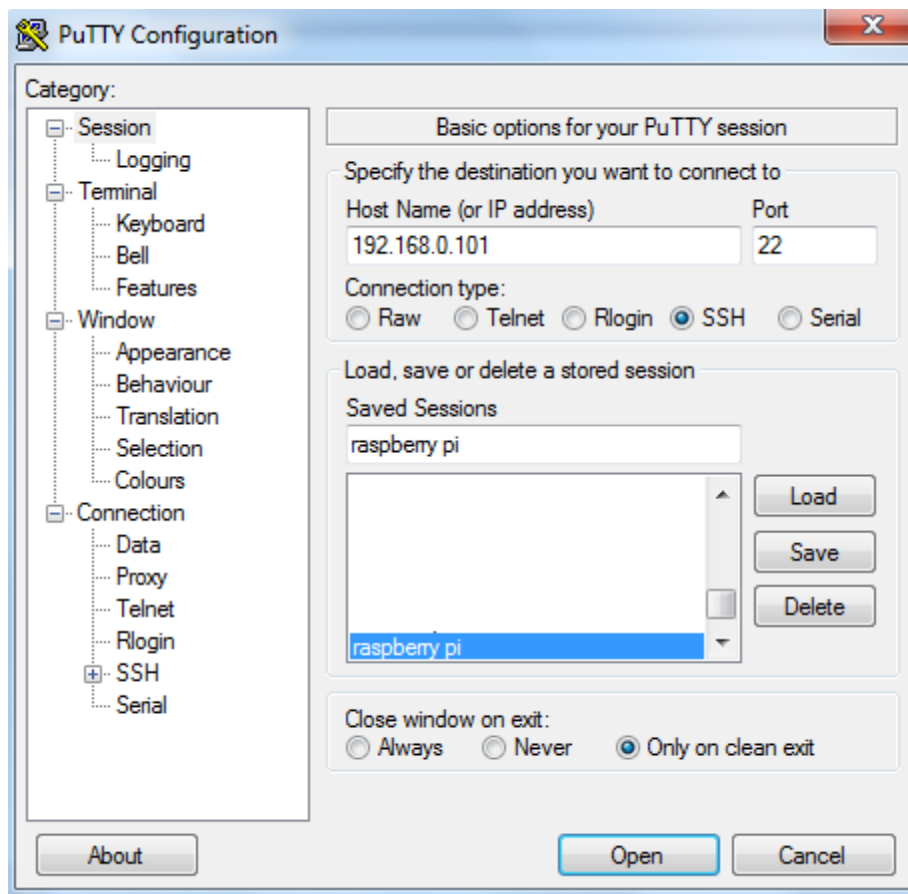


Slika 3.2: Prikaz grafičnega vmesnika operacijskega sistema Raspbian.

3.1.3 Oddaljen dostop


S pomočjo protokola in orodja za varno povezavo SSH (angl. secure shell) smo omogočili dostop iz drugega računalnika do našega računalnika RPI v istem omrežju [12]. Za uspešno povezavo je bilo potrebno vnesti le naslov naprave IP (angl. internet protocol) ter uporabniško ime in geslo. Do naslova IP naprave pridemo tako, da v konzoli vpišemo ukaz *ifconfig*. Privzeto uporabniško ime je *pi* geslo pa *raspberrypi*.

Za povezavo preko protokola SSH smo uporabili brezplačni program PuTTY (Slika 3.3), katerega je mogoče prenesti iz uradne spletne strani [13] projekta PuTTY.



Slika 3.3: Prikaz prijave v okolje SSH znotraj programa PuTTY.

Tako lahko upravljamo z računalnikom RPI, brez da bi nanj povezali kakršnokoli vhodno oziroma izhodno napravo kot je monitor, tipkovnica, miška ali kaj podobnega (Slika 3.4).



```

pi@raspberrypi: ~/diploma
login as: pi
pi@192.168.0.101's password:
Linux raspberrypi 3.6.11-rpi-aufs #1 PREEMPT Sun Aug 11 20:06:52 CEST 2013 armv6
l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Sep  6 19:54:37 2014 from 192.168.0.100
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:06:75:94
          inet addr:192.168.0.101  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:210 errors:0 dropped:0 overruns:0 frame:0
          TX packets:86 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17392 (16.9 KiB)  TX bytes:12059 (11.7 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

pi@raspberrypi ~ $ cd /home/pi/diploma
pi@raspberrypi ~/diploma $

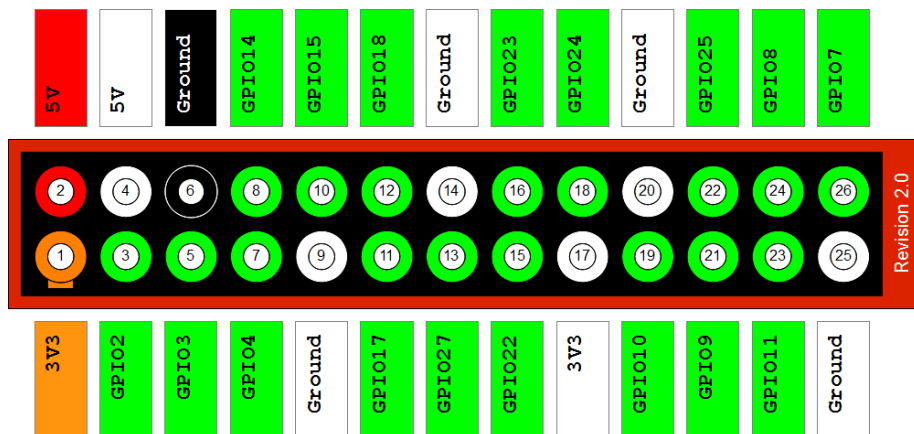
```

Slika 3.4: Prikaz uporabe varnega okolja SSH in izvajanje ukazov preko oddaljenega dostopa znotraj operacijskega sistema Windows.

3.1.4 Upravljanje s pini GPIO

Za branje in upravljanje stanj senzorjev ter drugih komponent, računalnik RPI uporablja digitalne pine GPIO (slika 3.5). Pinov GPIO je kar 26, razlikujejo pa se po namenu in načinu delovanja.

Za še enostavnejše upravljanje s pini GPIO smo namestili knjižnico *wiringPi* [14], katero je mogoče namestiti preko okolja SSH in git repozitorija z uporabo ukaza `git clone git://git.drogon.net/wiringPi`. Oznake pinov ter način in stanja pinov smo pregledovali z uporabo ukaza `gpio readall` (Slika 3.6).



Slika 3.5: Prikaz oznak in pomena pinov GPIO računalnika RPI.

```
$ gpio readall

+-----+--Rev2--+-----+-----+-----+
| wiringPi | GPIO | Phys | Name  | Mode | Value |
+-----+-----+-----+-----+-----+
| 0 | 17 | 11 | GPIO 0 | IN | Low |
| 1 | 18 | 12 | GPIO 1 | IN | Low |
| 2 | 27 | 13 | GPIO 2 | IN | Low |
| 3 | 22 | 15 | GPIO 3 | IN | Low |
| 4 | 23 | 16 | GPIO 4 | IN | Low |
| 5 | 24 | 18 | GPIO 5 | IN | Low |
| 6 | 25 | 22 | GPIO 6 | OUT | High |
| 7 | 4 | 7 | GPIO 7 | IN | Low |
| 8 | 2 | 3 | SDA | IN | High |
| 9 | 3 | 5 | SCL | IN | High |
| 10 | 8 | 24 | CE0 | ALT0 | High |
| 11 | 7 | 26 | CE1 | ALT0 | High |
| 12 | 10 | 19 | MOSI | ALT0 | Low |
| 13 | 9 | 21 | MISO | ALT0 | Low |
| 14 | 11 | 23 | SCLK | ALT0 | Low |
| 15 | 14 | 8 | TxD | ALT0 | High |
| 16 | 15 | 10 | RxD | ALT0 | High |
| 17 | 28 | 3 | GPIO 8 | IN | Low |
| 18 | 29 | 4 | GPIO 9 | IN | Low |
| 19 | 30 | 5 | GPIO10 | IN | Low |
| 20 | 31 | 6 | GPIO11 | IN | Low |
+-----+-----+-----+-----+-----+
```

Slika 3.6: Prikaz izpisa ukaza *gpio readall*.

Za potrebe izdelave našega sistema smo morali spremeniti zgolj način delovanja in stanja posameznega pina [12], [13]. Način delovanja smo spreminjali z uporabo ukaza *gpio mode* <številka pina> *in* ter *gpio mode* <številka pina> *out*. Stanja posameznega pina, ki deluje v načinu *out* pa smo nastavljali z uporabo ukaza *gpio write* <številka pina> 0 oziroma *gpio write* <številka pina> 1.

3.1.5 Programiranje v programskem jeziku Python

Python je visoko nivojski programski jezik, ki je že nameščen na operacijskem sistemu *Raspbian*. Prav tako je nameščen tudi urejevalnik kode *IDLE* za delo v jeziku Python. Zaradi njegove velike vloge, je drugi del imena računalnika Raspberry Pi dobil prav po njem [9].

3.2 Predstavitev izdelane rešitve

Preden smo se lotili pisanja programske rešitve v jeziku Python, smo upravljali pine GPIO kar preko konzole z uporabo knjižnice *wiringPi*. Ker pa nam dolgoročno ta rešitev ni ustrezala, smo razvili program, ki je vsakih nekaj sekund preverjal stanja naprav. Na podlagi pravil in stanj, shranjenih v podatkovni bazi, pa je upravljal z napravami.

3.2.1 Uporabljene knjižnice

Pri razvoju programske rešitve v jeziku Python smo si pomagali z uporabo različnih knjižnic. Za delo z branjem trenutne temperature iz temperaturnega senzorja, smo uporabili knjižnico *os* [15] ter *glob* [16]. Krmiljenje pinov GPIO smo si olajšali z uporabo knjižnice *RPi.GPIO* [17]. *Json* [18] in *urllib2* [19] smo uporabili za klice metod GET in POST ter branje vrnjenih rezultatov metod spletnega vmesnika. V primeru težav z vzpostavljanjem povezave s strežnikom, se sistem upravlja preko pravil, zapisanih v dokumentih tipa CSV (angl. comma separated value). V ta namen smo uporabili tudi knjižnico *csv* [20]. Za krmiljenje časovnega premora in pridobivanje trenutnega časa, pa smo uporabili knjižnico *time* [21] in *datetime* [22].

3.2.2 Krmiljenje temperature

Trenutno temperaturo v prostoru smo odčitali s pomočjo temperaturnega senzorja DS18B20. Senzor deluje tako, da temperaturo zapisuje v datoteko na naslovu */sys/bus/w1/devices/*, kjer ustvari dokument z imenom, ki predstavlja unikatno številko izbranega senzorja [23]. V našem primeru je celotna pot do dokumenta */sys/bus/w1/devices/28-0014151ba2ff*.

Napisali smo metodo *readTemperature*, ki vrne trenutno temperaturo v stopinjah Celzija, zaokroženo na dve decimalni mesti natančno. Celotno vsebino dokumenta, v katerega piše temperaturni senzor, smo prebrali s pomočjo metode *readTemperatureSensorFile*.

Vrnjeno trenutno temperaturo smo uporabili pri določanju trenutnega stanja električnega grelca in ventilatorja. Stanja smo nastavili v odvisnosti od pravila, ki določa idealno temperaturo v prostoru za trenutno starost piščancev.

3.2.3 Povezava s spletnim vmesnikom

Za povezavo s spletnim vmesnikom smo napisali vrsto metod, ki so nam omogočile vnos trenutnih stanj naprav v podatkovno bazo in branje shranjenih vrednosti iz podatkovne baze (Tabela 3.1). Metode se izvajajo na podlagi določenega naslova URL (angl. uniform resource locator), ki predstavlja osnovno pot do spletnega vmesnika.

Naziv metode	Opis metode
<i>getRaspberryConfig</i>	Prejme argument naziva polja, za katerega želimo izpisati trenutno vrednost, shranjeno v podatkovni bazi tabele <i>raspberry_config</i> . Nato se izvede klic metode spletnega vmesnika, ki vrne trenutno vrednost polja v podatkovni bazi.
<i>getAllRaspberryConfig</i>	Vrne vse trenutno shranjene vrednosti tabele <i>raspberry_config</i> .
<i>setRaspberryConfig</i>	Prejme argument naziva polja in vrednost, katerega nato tudi nastavi in vrne v rezultatu.
<i>getLightRule</i>	Prejme argument starosti piščancev v dnevih in trenutno uro, vrne pa vrednost stanja luči, ki jo določa vnaprej predpisano pravilo.
<i>setLightRule</i>	Prejme argument starosti piščancev v dnevih, trenutno uro in stanje luči. Te vrednosti vrne v rezultatu in jih shrani v tabelo <i>raspberry_light</i> .
<i>getCurrentLightRule</i>	Vrne vrednosti stanj pravila luči za trenutno starost piščancev in trenutno uro.
<i>getTemperatureRule</i>	Prejme argument starosti piščancev v dnevih. Vrne najnižjo in najvišjo idealno temperaturo za posredovano starost.

<i>setTemperatureRule</i>	Prejme argument starosti piščancev v dnevih ter najnižjo in najvišjo temperaturo. Te vrednosti vrne v rezultatu in nastavi v podatkovni bazi tabele <i>raspberry_temperature</i> .
<i>getCurrentTemperatureRule</i>	Vrne najnižjo ter najvišjo dovoljeno temperaturo za trenutno starost piščancev.
<i>getChickensAge</i>	Glede na trenutni cikel, posredovan kot argument metode, izračuna in vrne trenutno starost piščancev.

Tabela 3.1: Opis uporabljenih metod za povezavo spletnega vmesnika in programske rešitve v jeziku Python.

3.2.4 Upravljanje stanj pinov GPIO

Za upravljanje s stanji pinov GPIO smo napisali metodi *setPinValue* in *getPinValue*.

Metoda *setPinValue* prejme številko pina GPIO ter njegovo vrednost stanja, nato pa po potrebi spremeni trenutno nastavljeno vrednost stanja. Poleg tega tudi vrne obvestilo o spremembi stanja, v kolikor je prišlo do spremembe.

Metoda *getPinValue* pa zgolj prebere in vrne trenutno vrednost pina GPIO.

3.2.5 Glavna zanka programa

Celoten proces, ki se izvede ob zagonu našega programa, se lahko zelo razlikuje. Odvisen je od trenutnih shranjenih stanj.

Po vključitvi vseh uporabljenih knjižnic se najprej nastavijo potrebne nastavitve za branje temperature temperaturnega senzorja. Nato se vzpostavi povezava s spletnim vmesnikom. V primeru, da je spletni vmesnik dostopen, pa se te vrednosti tudi shranijo v podatkovno bazo preko metode *setRaspberryConfig*. Nastavijo se tudi načini delovanja stanj pinov GPIO. Vse našteto se izvede zgolj enkrat ob zagonu programa.

Poleg dela, ki se izvede zgolj enkrat ob zagonu programa, smo napisali tudi glavno zanko programa. Ta se izvaja stalno s premorom desetih sekund med zaključkom prejšnje iteracije in začetkom nove iteracije. V vsaki posamezni iteraciji se najprej izvede izpis trenutnega časa, nato pa se preveri nastavljeno stanje naše naprave. To stanje določa vrednost polja *status* v podatkovni bazi tabele *raspberry_config*, pomeni pa vklop oziroma izklop krmiljenja. Kasneje se prebere trenutna temperatura ter stanja zgornjega in spodnjega senzorja nivoja

vode, ki sta nameščena v zbiralniku za vodo. V spremenljivke se shranijo še prejšnja nastavljena stanja naprav, nato pa se izvede preverjanje in nastavljanje trenutnih stanj naprav. Krmiljenje je mogoče na dva različna načina, in sicer preko spletnega vmesnika z metodo *onlineMainLoop* ter brez povezave s spletnim vmesnikom z metodo *offlineMainLoop*.

V načinu krmiljenja s spletnim vmesnikom, se najprej v podatkovno bazo tabele *raspberry_config* shrani trenutna temperatura ter stanja zgornjega in spodnjega senzorja nivoja vode. Nato se shranijo še vsa trenutna stanja tabele *raspberry_config*. Te trenutno shranjene nastavitve se kasneje pošljejo kot argument metod za krmiljenje naprav, saj je končna odločitev o spremembi stanj odvisna prav od trenutnih nastavitvev in pravil. Metode za krmiljenje naprav preverijo in nastavijo stanje luči, temperaturno odvisnih naprav in upravljanje točenja vode, nato pa še zapišejo vsa trenutna stanja tabele *raspberry_config* v tabelo *raspberry_statistics* (Tabela 3.2).

Naziv metode	Opis metode
<i>checkLight</i>	Prejme argument z vrednostmi trenutnih nastavitvev. Nato na podlagi pravil in trenutnega stanja po potrebi spremeni stanje pina GPIO, ki je namenjen upravljanju luči.
<i>checkTemperature</i>	Prejme argument z vrednostmi trenutnih nastavitvev. Nato na podlagi pravil in trenutnega stanja po potrebi spremeni stanje pina GPIO, ki je namenjen upravljanju električnega grelca in električnega ventilatorja.
<i>checkWater</i>	Prejme argument z vrednostmi trenutnih nastavitvev. Nato na podlagi pravil in trenutnega stanja senzorjev nivoja vode v zbiralniku po potrebi spremeni stanje pina GPIO, ki upravlja točenje vode.
<i>logStatistics</i>	Beleži trenutna stanja v tabelo <i>raspberry_statistics</i> .

Tabela 3.2: Predstavitev metod, ki se izvedejo znotraj metode *onlineMainLoop*.

Krmiljenje naprav z metodo *offlineMainLoop* se izvaja takrat, ko spletni vmesnik ni na voljo. Lahko se namreč zgodi, da pride do izpada spletnega strežnika ali podobnih težav, mi pa si ne smemo dovoliti, da takrat naš sistem ne bi deloval. V ta namen smo napisali metode, ki krmilijo z napravami preko zapisov dokumentov tipa CSV, brez pomoči spletnega vmesnika.

S klicem metode *offlineMainLoop* se najprej izvede izračun starosti piščancev, preko dokumenta CSV, nato pa se izvedejo še preostale metode za krmiljenje stanj pinov GPIO.

Naziv metode	Opis metode
<i>checkLightOffline</i>	Prejme argument starosti piščancev, trenutno uro in stanje pina GPIO, ki skrbi za krmiljenje luči. Nato na podlagi vnaprej predpisanih pravil po potrebi spremeni stanje pina GPIO.
<i>checkTemperatureOffline</i>	Prejme argument starosti piščancev, trenutne temperature in trenutnega stanja električnega grelca ter ventilatorja. Nato na podlagi vnaprej predpisanih pravil in trenutnih stanj po potrebi spremeni stanje pina GPIO.
<i>checkWaterOffline</i>	Prejme argument stanja senzorjev nivoja vode v zbiralniku ter trenutno stanje naprave za točenje vode. Nato na podlagi pravil in trenutnega stanja po potrebi spremeni stanje pina GPIO.
<i>logStatisticsOffline</i>	Metoda prejme argument trenutnega cikla in starost piščancev. Nato zapiše trenutna stanja v dokument statistics.csv.

Tabela 3.3: Predstavitev metod, ki se izvedejo znotraj metode *offlineMainLoop*.

Poglavje 4 Elektronsko vezje naprav

Za krmiljenje našega sistema smo načrtali vezje naprav, ki se odziva glede na stanja pinov GPIO računalnika RPI. Glede na naše potrebe in zmogljivosti sistema, smo izbrali ustrezne komponente in jih združili v prenosljivo vezje naprav. Načrtovanje vezja smo se lotili postopoma. Za vsako posamezno komponento smo preverili njihove lastnosti, zahteve in sposobnosti.

4.1 Predstavitev uporabljenih tehnologij, orodij in naprav

Za izdelano rešitev smo potrebovali ohišje primernih dimenzij, napajanje za komponente, relejske module za preklon med stanji naprav, temperaturni senzor, senzorja nivoja vode, elektromagnetni ventil ter ostale osnovne komponente za povezavo vezja v delujoč električni krog.

4.1.1 Ohišje

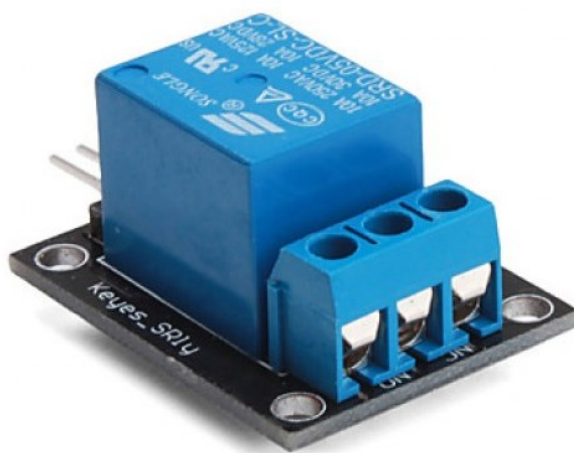
Vezje smo postavili v ognjevarno ohišje, ki nudi ustrezno zaščito pred nezaželenimi zunanji vplivi. V njem je dovolj prostora za porazdelitev vseh uporabljenih komponent, poleg tega pa omogoča tudi prenosljivost sistema. Dodatna predelava ohišja ni bila potrebna, saj že vsebuje devet lukenj z uvodnicami. Skoznje smo napeljali napajalni kabel, kabel za senzorja nivoja vode in temperature ter elektromagnetni ventil.

4.1.2 Napajanje

Ker je za krmiljenje elektromagnetnega ventila potrebna napetost 12 V, za večino preostalih komponent pa 5 V, smo za napajanje sistema izbrali računalniški napajalni kabel, ki že vsebuje 5 V in 12 V napetost. 5 V napetost smo uporabili za napajanje računalnika RPI, relejske module ter senzorje nivoja vode. 12 V napetost smo uporabili za krmiljenje elektromagnetnega ventila.

4.1.3 Relejski modul Keyes_SRly

Možnost upravljanja stanj naprav smo pridobili z uporabo relejskih modulov *Keyes_SRly* [24] (Slika 4.1). Modul že vsebuje tranzistor, ki skrbi za preklop relejske tuljave in kontrolno diodo LED (angl. light emitting diode) relejskega modula, ki sveti v primeru visokega stanja izhodne napetosti.

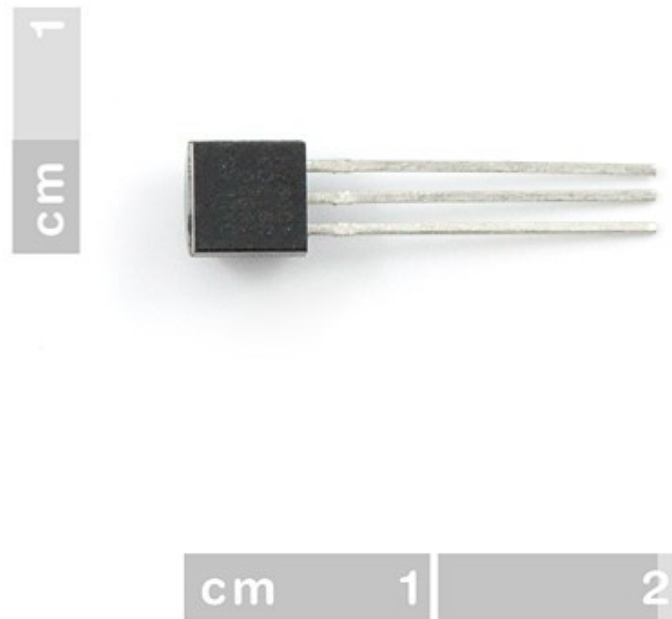


Slika 4.1: Prikaz relejskega modula *Keyes_SRly* [25].

Za delovanje modula skrbijo tri vhodne nožice, namenjene 5 V napajanju ter ozemljitvenem in krmilnem signalu. Ker za krmilni signal zadostuje že 3.3 V napetost, smo le-tega lahko nastavljali kar s stanji pinov GPIO računalnika RPI. Na izhodni strani modula pa velja omeniti predvsem privzeto odprt signal NO (angl. normaly open) in privzeto zaprt signal NC (angl. normaly closed), ki ju uporabimo glede na potrebe naprave, ki jo krmilimo [26].

4.1.4 Senzor temperature DS18B20

Za merjenje temperature smo uporabili senzor *DS18B20*, ki je zelo majhen in povsem primeren za naše potrebe (Slika 4.3). Senzor je dimenzije 0,4 x 1,9 cm. Napajalna napetost senzorja je od 3 do 5,5 V [27]. Omogoča merjenje temperature v območju od -55 °C do +125 °C z natančnostjo ± 0.5 °C v območju od -10 °C do +85 °C. Vsak senzor *DS18B20* ima določeno unikatno 64-bitno serijsko kodo, shranjeno v spominu ROM [4].



Slika 4.3: Prikaz velikosti senzorja DS18B20 [28].

4.1.5 Senzor nivoja vode

Za nadzor nivoja vode v zbiralniku smo uporabili dva horizontalna senzorja tekočine (Slika 4.4). Senzor deluje kot stikalo med visokim in nizkim signalom. Izhodni signal je odvisen od trenutnega nivoja vode. Ko nivo vode pade, je plavajoči del senzorja na najnižji točki, ko pa nivo vode naraste, pa plavajoči del senzorja doseže najvišjo točko in s tem tudi spremeni izhodni signal.



Slika 4.4: Prikaz senzorja nivoja vode [29].

Izhodni signal lahko določimo z ustrezno postavitvijo senzorja. Če želimo, da sta izhodni stanji ravno obratni glede na nivo vode, senzor le obrnemo za 180 °. Temperaturni razpon, v katerem senzor deluje, je od -20 °C do +80 °C. Najvišja preklopna napetost je 100 V [30].

4.1.6 Elektromagnetni ventil TFW-1S

Za točenje vode smo uporabili elektromagnetni ventil za vodo *TFW-1S* (Slika 4.5). Ventil je privzeto zaprt, kar pomeni, da omogoča pretok vode le ob visokem signalu. Za napajanje potrebuje 12 V napetost [31]. Izhodni in vhodni nastavek za dotok in odtok vode je primeren za 1/2" cevi.



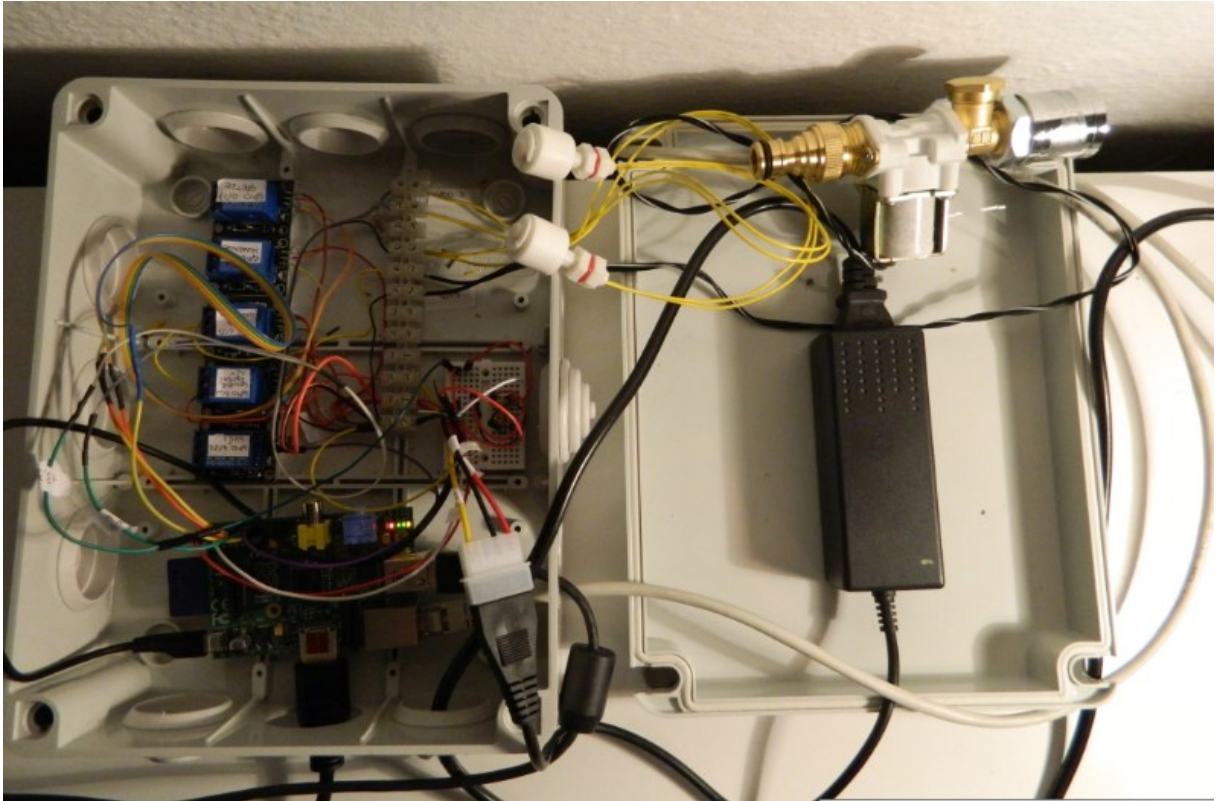
Slika 4.5: Prikaz elektromagnetnega ventila za vodo TFW-1S [32].

4.2 Predstavitev izdelane rešitve

Izdelovanja elektronskega vezja (Slika 4.6) smo se lotili postopoma. Komponente vezja smo najprej preučevali posamezno, ko pa smo spoznali delovanje, pa smo vse skupaj povezali v celoto.

Vezje je postavljeno v ognjevarno ohišje, ki omogoča prenosljivost in zaščito pred zunanjimi vplivi. Za vir napetosti vezja in računalnika RPI skrbi računalniški napajalnik z napetostjo 5 V in 12 V. Preko vhodnih in izhodnih pinov GPIO računalnika RPI se upravljajo relejski moduli, ki služijo krmiljenju naprav. Za krmiljenje osvetlitve prostora zadostuje zgolj kombinacija relejskega modula in krmilnega signala pina GPIO. Temperatura prostora se upravlja na podlagi dveh različnih relejskih modulov. Eden za hlajenje, drugi pa za ogrevanje prostora. Poleg tega je za krmiljenje temperature potrebna informacija temperaturnega senzorja DS18B20. Za krmiljenje nivoja vode v zbiralniku služita dva senzorja oziroma

nivojski stikali in elektromagnetni ventil, ki za svoje delovanje potrebuje 12 V napetost. Zaradi varnosti smo na tem mestu poskrbeli tudi za malce bolj mehansko kontrolo nivoja vode, saj si nismo smeli privoščiti napak.

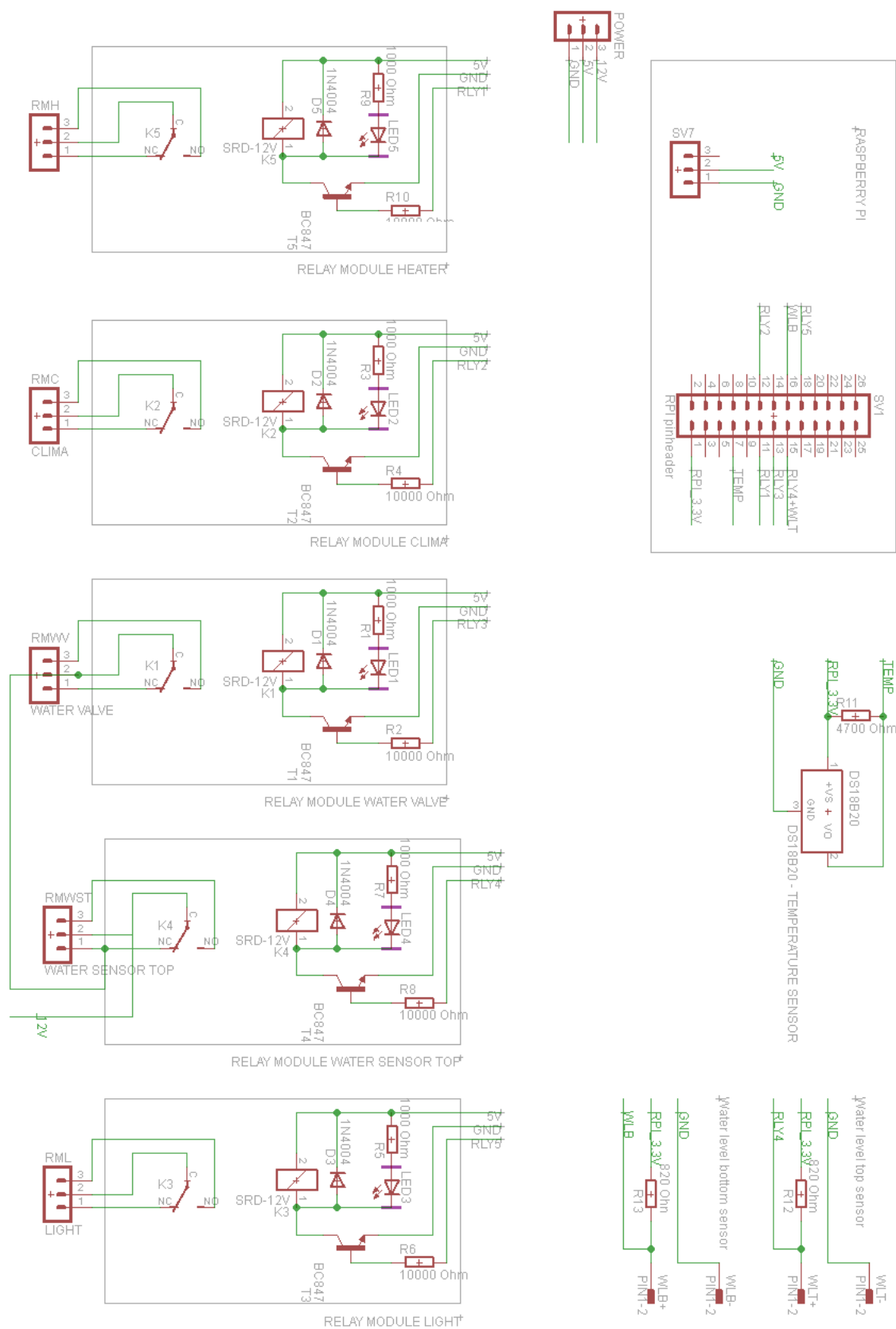


Slika 4.6: Prikaz elektronskega vezja naprav našega sistema.

4.2.1 Shema elektronskega vezja

Izdelali smo shemo elektronskega vezja (Slika 4.7), ki prikazuje povezavo naših uporabljenih komponent. Na shemi ni prikazan priklop končnih naprav kot so električni grelec, električni ventilator, senzorja nivoja vode, elektromagnetni ventil in luč.

Električni grelec je možno povezati na pine RMH, električni ventilator na pine RMC, elektromagnetni ventil na pine RMWV ter luč na pine RML. Vsem tem napravam je skupno to, da napajalno napetost povežemo preko pina 2 (C) posameznega relejskega modula, nato pa se odločimo kakšno vlogo naj imajo naše naprave. Če želimo da je naprava privzeto vključena, potem povežemo s posamezno napravo še pin 1 (NC), sicer pa pin 3 (NO). Za povezavo senzorjev nivoja vode služijo štirje pini, in sicer WLT- in WLT+ za zgornji senzor nivoja vode, pina WLB- in WLB+ pa za spodnji senzor nivoja vode.



Slika 4.7: Prikaz sheme elektronskega vezja.

Poglavje 5 Spletni vmesnik

Vzpostavili smo sistem podatkovne baze, spletnega vmesnika API ter grafičnega vmesnika v obliki nadzorne plošče, ki prijavljenim uporabnikom dovoljuje upravljanje s stanji naprav. S klikom na določeno akcijo v nadzorni plošči se izvede klic vmesnika, ki posodobi oziroma izpiše vrednosti stanj shranjenih v podatkovni bazi.

5.1 Predstavitev uporabljenih tehnologij

Uporabili smo sistem za upravljanje podatkovnih baz, ter ostale zelo pogosto uporabljene spletne tehnologije za izdelavo spletnega vmesnika API in nadzorne plošče sistema.

5.1.1 MySQL

MySQL je sistem za upravljanje s podatkovnimi bazami, ki je ime dobilo po hčerki ustanovitelja z imenom My ter jezika SQL (angl. structured query language) [33]. Gre za odprtokodno implementacijo relacijske podatkovne baze, ki za delo s podatki uporablja jezik SQL. Deluje na principu odjemalec – strežnik. Napisan je v jeziku C in C++. MySQL je najbolj uporabljen odprtokodni sistem za upravljanje s SQL podatkovnimi bazami.

5.1.2 HTML

HTML (angl. hypertext markup language) je označevalni jezik za izdelavo spletnih strani [33]. Koda je zgrajena iz značk, uporabljenih v parih, kot začetna in zaključna značka. Določajo strukturo ter vsebino spletnih strani. Z njim je mogoče definirati več različnih elementov kot so tabele, fotografije, slike, povezave, besedila in podobno.

5.1.3 CSS

CSS (angl. cascading style sheets) je jezik za določanje oblike posameznih elementov spletne strani [33]. Omogoča nam, da obliko povsem ločimo od strukture in vsebine spletne strani. Z njim lahko določamo barvo, odmik, obrobe ter podobne oblikovne lastnosti. Prav tako nam omogoča spremembo oblike ob različnimi aktivnostih (na primer prekritje miške čez posamezen element, predhoden klik na element povezave in podobno). Poleg jezika HTML je

to osnova za izdelavo spletne strani. Pri izdelavi naše rešitve smo uporabili različico jezika CSS3.

5.1.4 PHP

PHP (angl. hypertext preprocessor) je razširjen odprtokodni programski jezik, ki se uporablja za razvoj dinamičnih spletnih strani [33]. Teče na spletnem strežniku, kjer PHP kodo pretvori v dinamično spletno stran. Lahko je vključen v kodo HTML jezika.

5.1.5 Javascript

Javascript je objektni programski jezik za ustvarjanje interaktivnih spletnih strani [33]. V sodelovanju z jezikom HTML poživi stran z dinamičnim izvajanjem. Je odprtokodni jezik, podobne sintakse kot jezik C. Javascript metode se izvedejo ob sprožitvi nekega dogodka (na primer klik miške, premik miške, klik na tipko tipkovnice in podobno). Izvaja se na strani odjemalca. Spremembe, ki jih povzroči Javascript metoda, so vidne takoj, saj ne potrebujejo osveževanja spletne strani.

5.1.6 jQuery

Uporabili smo knjižnico jQuery skriptnega jezika Javascript. Glavni namen uporabe te knjižnice je, da z manj napisane kode naredimo več [34]. Uporabili smo jo predvsem zaradi enostavne uporabe in potrebe po dodajanju raznih učinkov in animacij naše nadzorne plošče.

5.1.7 JSON

JSON (angl. javascript object notation) je odprtokodni format za izmenjavo podatkov [35]. Temelji na uporabi jezika Javascript. Uporabljen je kot alternativa jezika XML (angl. extensible markup language). Podatki so predstavljeni kot objekti neurejenih množic parov imena in vrednosti. Začnejo se z zavitim oklepajem, končajo pa z zavitim zaklepajem. Vsakemu imenu sledi dvopičje, dvopičju pa vrednost. Posamezni pari so med seboj ločeni z vejicami.

5.1.8 Ajax

Ajax (angl. asynchronous javascript + XML) je skupek tehnik spletnega razvoja, ki tečejo na strani odjemalca [36]. Uporablja se za asinhrono pošiljanje in prejemanje informacij v ozadju brez potrebe po ponovnem nalaganju spletne strani. Služi za izdelavo interaktivnih spletnih strani.

5.1.9 Bootstrap

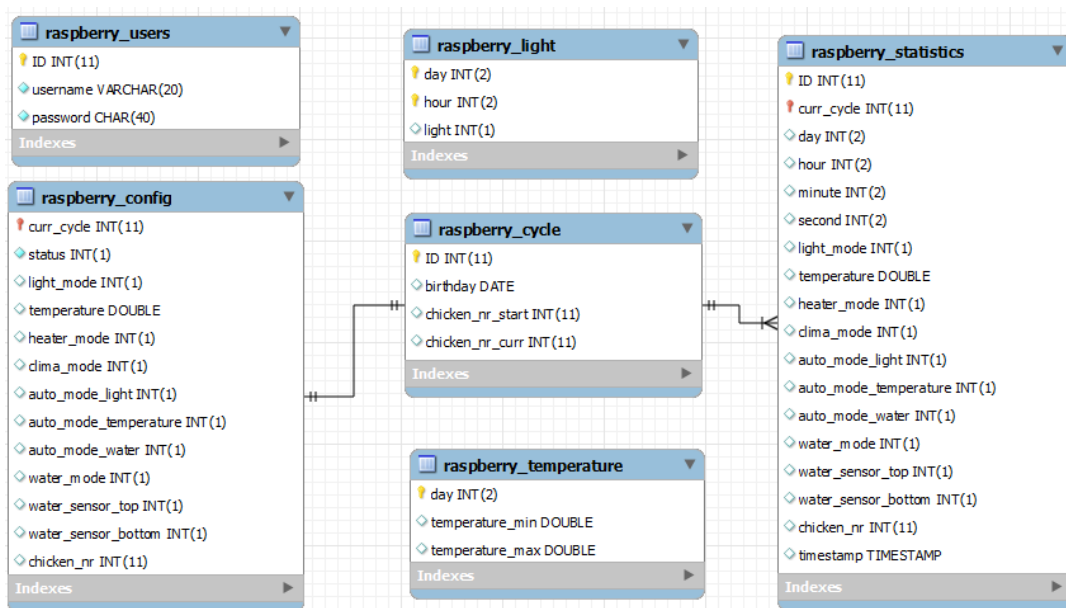
Bootstrap je brezplačna zbirka orodij za ustvarjanje spletnih strani in aplikacij. Vsebuje tehnologije HTML, CSS in Javascript [37]. Zaradi že izdelanih rešitev za uporabo gumbov, obvestil, zavihkov in ostalih elementov spletne strani, poskrbi za hitrejši razvoj spletnih aplikacij. Omogoča razvoj odzivnih spletnih strani z uporabo sistema mreže dvanajstih stolpcev, ki se prikazujejo različno, glede na dimenzijo naprave.

5.2 Predstavitev izdelane rešitve

Za shranjevanje podatkov trenutnih stanj naprav smo uporabili sistem za upravljanje s podatkovnimi bazami MySQL. Izdelali smo vmesnik API, s pomočjo predloge *Slim* [38], za lažje delo s shranjenimi podatki. Razvili smo tudi nadzorno ploščo sistema, z uporabo predloge *AdminLTE* [39], za lepši pregled in boljši nadzor nad trenutnimi shranjenimi stanji. Pri obeh uporabljenih predlogah smo sistem dodatno prilagodili svojim potrebam.

5.2.1 Podatkovna baza

Za upravljanje s podatkovno bazo smo uporabili tehnologijo MySQL. Znotraj podatkovne baze smo ustvarili šest tabel in sicer *raspberry_config*, *raspberry_cycle*, *raspberry_light*, *raspberry_temperature*, *raspberry_statistics* ter *raspberry_users* (Slika 5.1). Tabela *raspberry_users* skrbi za potrebe prijavnega obrazca nadzorne plošče, vse ostale tabele pa so namenjene neposrednemu krmiljenju sistema zgrajenega elektronskega vezja.



Slika 5.1: Prikaz diagrama in relacij tabel podatkovne baze.

5.2.1.1 Tabela raspberry_config

V tabeli *raspberry_config* so shranjena trenutna stanja naprav. S pomočjo te tabele se krmilijo vse naprave našega sistema (Tabela 5.1).

Ime polja	Tip	Pomen vrednosti
<i>status</i>	int (1)	0 – izklop krmiljenja naprav. 1 – vklop krmiljenja naprav.
<i>curr_cycle</i>	int (11)	Ident trenutnega cikla piščancev.
<i>light_mode</i>	int (1)	0 – izklop luči. 1 – vklop luči.
<i>temperature</i>	double	Trenutno izmerjena temperatura v prostoru, zapisana kot decimalna vrednost. Enoto temperature (°C), zaradi večje preglednosti, nismo shranjevali v polje temperature.
<i>heater_mode</i>	int (1)	0 – izklop električnega grelca. 1 – vklop električnega grelca.
<i>clima_mode</i>	int (1)	0 – izklop ventilatorja. 1 – vklop ventilatorja.
<i>auto_mode_light</i>	int (1)	0 – izklop samodejnega krmiljenja svetlobe. 1 – vklop samodejnega krmiljenja svetlobe.
<i>auto_mode_temperature</i>	int (1)	0 – izklop samodejnega krmiljenja temperature. 1 – vklop samodejnega krmiljenja temperature.
<i>auto_mode_water</i>	int (1)	0 – izklop samodejnega krmiljenja vode. 1 – vklop samodejnega krmiljenja vode.
<i>water_mode</i>	int (1)	0 – izklop točenja vode. 1 – vklop točenja vode.
<i>water_sensor_top</i>	int (1)	0 – nivo vode ne dosega zgornjega senzorja. 1 – nivo vode dosega zgornji senzor.

<i>water_sensor_bottom</i>	int (1)	0 – nivo vode ne dosega spodnjega senzorja. 1 – nivo vode dosega spodnji senzor.
<i>chicken_nr</i>	int (11)	Trenutno število piščancev.

Tabela 5.1: Predstavitev polj tabele *raspberry_config*.

5.2.1.2 Tabela *raspberry_cycle*

V tabelo *raspberry_cycle* se shranjuje informacija o trenutnem ciklu piščancev (Tabela 5.2). Pomembna je predvsem zato, da lahko izračunamo starost trenutnega cikla piščancev in beležimo umrljivost. Primarni ključ te tabele je polje *ID*, ki se ob novem vnosu vrednosti samodejno povečuje.

Ime polja	Tip	Pomen vrednosti
<i>ID</i>	int (11)	Zaporedna številka cikla piščancev.
<i>birthday</i>	date	Dan izvalitve piščancev.
<i>chicken_nr_start</i>	int (11)	Začetno število piščancev.
<i>chicken_nr_curr</i>	Int(11)	Zadnje zabeleženo število piščancev.

Tabela 5.2: Predstavitev polj tabele *raspberry_cycle*.

5.2.1.3 Tabela *raspberry_light*

V tabeli *raspberry_light* je opisano pravilo prižiganja luči v odvisnosti od starosti piščancev in trenutne ure. Primarna ključa te tabele sta polji *day* in *hour*. V kolikor bi želeli določiti, da mora biti prižgana luč pri štiri dni starih piščancih ob 17. uri, moramo v polje *day* shranimo vrednost 4, v polje *hour* 17 ter v polje *light* 1.

Ime polja	Tip	Pomen vrednosti
<i>day</i>	int (2)	Starost piščancev v dnevih.
<i>hour</i>	int(2)	Ura v dnevu.

<i>light</i>	int (1)	0 - luči morajo biti ugasnjene. 1 - luči morajo biti prižgane.
--------------	---------	---

Tabela 5.3: Predstavitev polj tabele *raspberry_light*.

5.2.1.4 Tabela *raspberry_temperature*

V tabeli *raspberry_temperature* je opisano pravilo krmiljenja temperature v odvisnosti od starosti piščancev. Primarni ključ te tabele je polje *day*.

Ime polja	Tip	Pomen vrednosti
<i>day</i>	int (2)	Starost piščancev v dnevih.
<i>temperature_min</i>	double	Najnižja priporočena temperatura.
<i>temperature_max</i>	double	Najvišja priporočena temperatura.

Tabela 5.4: Predstavitev polj tabele *raspberry_temperature*.

5.2.1.5 Tabela *raspberry_statistics*

Za beleženje trenutnih stanj in pravil naših naprav smo izdelali tabelo *raspberry_statistics*. Služi nam predvsem za izdelavo poročil oziroma prikaz statističnih podatkov.

Ime polja	Tip	Pomen vrednosti
<i>ID</i>	int (11)	Zaporedna številka zapisa.
<i>curr_cycle</i>	int(11)	Nastavljen cikel piščancev.
<i>day</i>	int (2)	Starost piščancev v dnevih.
<i>hour</i>	int (2)	Ura zapisa.
<i>minute</i>	int (2)	Minuta zapisa.
<i>second</i>	int (2)	Sekunda zapisa.
<i>light_mode</i>	int (1)	Stanje luči.

<i>temperature</i>	double	Trenutna temperatura.
<i>heater_mode</i>	int (1)	Stanje električnega grelca.
<i>clima_mode</i>	int (1)	Stanje ventilatorja.
<i>auto_mode_light</i>	int (1)	Nastavljeno samodejno oziroma ročno krmiljenje luči.
<i>auto_mode_temperature</i>	int (1)	Nastavljeno samodejno oziroma ročno krmiljenje temperature.
<i>auto_mode_water</i>	int (1)	Nastavljeno samodejno oz. ročno krmiljenje vode.
<i>water_mode</i>	int (1)	Stanje točenja vode.
<i>water_sensor_top</i>	int (1)	Stanje nivoja vode zgornjega senzorja.
<i>water_sensor_bottom</i>	int (1)	Stanje nivoja vode spodnjega senzorja.
<i>chicken_nr</i>	int (11)	Trenutno število piščancev v reji.
<i>timestamp</i>	timestamp	Čas zapisa.

Tabela 5.5: Predstavitev polj tabele *raspberry_statistics*.

5.2.1.6 Tabela *raspberry_users*

V tabeli *raspberry_users* se hranijo podatki o uporabnikih, ki imajo dostop do nadzorne plošče.

Ime polja	Tip	Pomen vrednosti
<i>ID</i>	int (11)	Zaporedna številka zapisa.
<i>username</i>	varchar (20)	Uporabniško ime uporabnika.
<i>password</i>	char (40)	Kriptirano geslo uporabnika.

Tabela 5.6: Predstavitev polj tabele *raspberry_users*.

5.2.2 Vmesnik za upravljanje s podatkovno bazo

Izdelali smo vmesnik API, ki preko GET in POST zahtev upravlja s podatki podatkovne baze. Uporabili smo predlogo *Slim*, ki je brezplačna in jo je mogoče prenesti iz uradne spletne strani predloge [38]. Namenjena je izdelavi preprostih spletnih aplikacij in vmesnikov API v programskem jeziku PHP.

5.2.2.1 Predstavitev metod vmesnika API

Metode se izvajajo na podlagi naslova URL, ki v določenih primerih vsebuje tudi argumente izvedenih metod. Za izpis zapisov podatkovne baze, smo napisali metode GET, za vnos novih oziroma posodabljanje obstoječih zapisov podatkovne baze, pa smo napisali metode POST (Tabela 5.7). Izvršujejo se ob zahtevah programa v jeziku Python, ki teče na računalniku RPI, ter znotraj sproženih akcij nadzorne plošče spletnega vmesnika.

Tip	Relativna pot URL	Opis metode
GET	<i>/raspberry/config/:config</i>	Vrne vrednost zapisa polja podatkovne baze, ki je posredovan kot argument (:config) znotraj naslova URL.
POST	<i>/raspberry/config/:config /:value</i>	Nastavi in vrne vrednost (:value) za polje (:config) podatkovne baze. Obe vrednosti sta zapisani znotraj naslova URL.
GET	<i>/raspberry/chickens_age/id /:id</i>	Vrne trenutno starost piščancev posredovanega identa (:id) cikla piščancev.
GET	<i>/raspberry/chickens_age /current</i>	Vrne trenutno starost piščancev za trenutno vnesen cikel piščancev.
GET	<i>/raspberry/light/:day/:hour</i>	Vrne pravilo stanja luči za posredovano starost piščancev (:day) in uro (:hour).
POST	<i>/raspberry/light/:day/:hour /:light</i>	Nastavi in vrne pravilo stanja luči (:light) za posredovano starost piščancev (:day) in uro (:hour).
GET	<i>/raspberry/temperature/:day</i>	Vrne temperaturno pravilo najnižje ter najvišje dovoljene temperature za posredovano starost piščancev (:day).

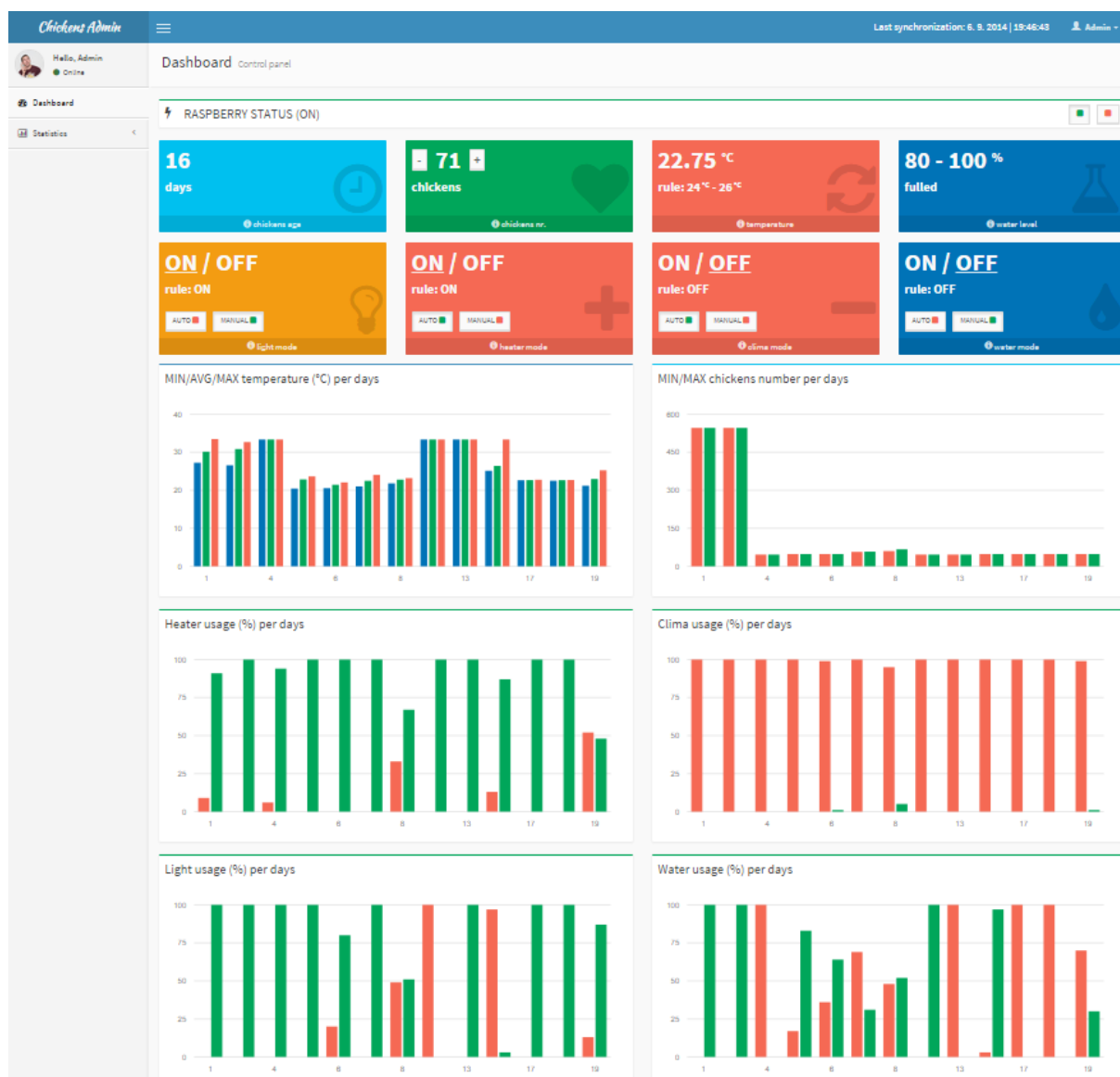
POST	<i>/raspberry/temperature/:day /:temperature_min/:temperature_max</i>	Nastavi in vrne pravilo posredovane najnižje (:temperature_min) in najvišje temperature (:temperature_max) za podano starost piščancev (:day).
GET	<i>/raspberry/rule/current/light</i>	Vrne informacije o pravilu stanja luči, za trenutno starost piščancev.
GET	<i>/raspberry/rule/current /temperature</i>	Vrne informacijo o temperaturnem pravilu, za trenutno starost piščancev.
POST	<i>/raspberry/number/:value</i>	Nastavi in vrne trenutno število piščancev, kateremu prišteje posredovano število (:value).
POST	<i>/raspberry/statistics</i>	Zabeleži in vrne trenutna stanja v tabelo, kjer se shranjujejo informacije o preteklih stanjih.
GET	<i>/raspberry/statistics/usage /:mode</i>	Vrne informacijo za posredovano stanje (:mode) iz tabele, kjer se hranijo podatki o preteklih stanjih.
GET	<i>/raspberry/statistics /temperature</i>	Vrne najnižjo, povprečno ter najvišjo temperaturo za posamezen dan.
GET	<i>/raspberry/statistics/number</i>	Vrne najvišje ter najnižje število piščancev za posamezen dan.
GET	<i>/raspberry/statistics /datetime</i>	Vrne datum in čas zadnje sinhronizacije krmiljenja s spletnim vmesnikom.
GET	<i>/raspberry/statistics /number-temperature</i>	Vrne najmanjše število piščancev posameznega dneva, prav tako pa tudi najnižjo ter najvišjo zabeleženo temperaturo.

Tabela 5.7: Predstavitev metod vmesnika API.

5.2.3 Nadzorna plošča

Izdelali smo nadzorno ploščo sistema, ki omogoča nadzor nad napravami preko spleta. Za upravljanje potrebujemo zgolj uporabniško ime in geslo, saj je dostop do nadzorne plošče dovoljen le registriranim uporabnikom.

Uporabili smo veliko različnih tehnologij, in sicer HTML, CSS, PHP, AJAX in Javascript, znotraj katere velja omeniti še uporabo knjižnice jQuery ter JSON. Delo smo si olajšali z uporabo predloge *AdminLTE*, katero smo prilagodili našim potrebam (Slika 5.2).



Slika 5.2: Prikaz nadzorne plošče našega sistema.

5.2.3.1 Uporaba predloge AdminLTE

Za izdelavo nadzorne plošče sistema smo uporabili odprtokodno predlogo *AdminLTE*. Dobiti jo je mogoče na uradni spletni strani predloge [39]. Zgradba predloge temelji na uporabi tehnologije Bootstrap3, HTML5 ter CSS3. Predloga že vsebuje pripravljeno obliko različnih elementov spletne strani kot so ikone, gumbi, obvestila, grafi in podobno. Implementirano ima tudi odzivno obliko, zato je prikaz nadzorne plošče pregleden tudi na ostalih napravah z nižjo resolucijo ekrana.

5.2.3.2 Izdelava obrazca za prijavo in odjavo

Za ogled nadzorne plošče je potrebna predhodna prijava z lastnim uporabniškim imenom in geslom. V kolikor uporabnik v podatkovni bazi oziroma natančneje v tabeli *raspberry_users* obstaja in je vneseno geslo pravilno, se v sejo shranijo podatki uporabnika, izvede pa se tudi preusmeritev na prvo stran nadzorne plošče. V primeru nepravilno vnesenih podatkov, pa se izvede preusmeritev na stran za ponovno prijavo. Ko je uporabnik prijavljen v nadzorno ploščo, je mogoča tudi odjava iz sistema. V ta namen smo razvili skripto, ki zgolj uniči sejo, uporabnika pa ponovno preusmeri na stran za prijavo.

5.2.3.3 Predelava uporabljene predloge

Z namestitvijo predloge *AdminLTE* smo pridobili nadzorno ploščo s statično vsebino. Ker to ni zadoščalo našim potrebam, smo napisali več skript, ki so omogočile prikaz podatkov ter možnost krmiljenja trenutnih stanj naprav.

Najprej smo napisali skripto, ki s klici metod vmesnika API poskrbi za izpis vseh trenutnih zapisov podatkovne baze in vrne rezultat v obliki JSON (Slika 5.3).

```
{"all_config":
  {"status":"1","curr_cycle":"2","light_mode":"1","temperature":"22.75",
    "heater_mode":"1","clima_mode":"0","auto_mode_light":"0",
    "auto_mode_temperature":"1","auto_mode_water":"0","water_mode":"0",
    "water_sensor_top":"1","water_sensor_bottom":"1","chicken_nr":"71"},
  "light_rule":{"day":15,"hour":"12","light":"1"},
  "temperature_rule":{"day":15,"temperature_min":"24","temperature_max":"26"},
  "chickens_age":{"chickens_age":15},
  "synchronization":{"date":"6. 9. 2014","time":"19:46:43"}}
```

Slika 5.3: Prikaz trenutnih zapisov vrnjenih v obliki JSON.

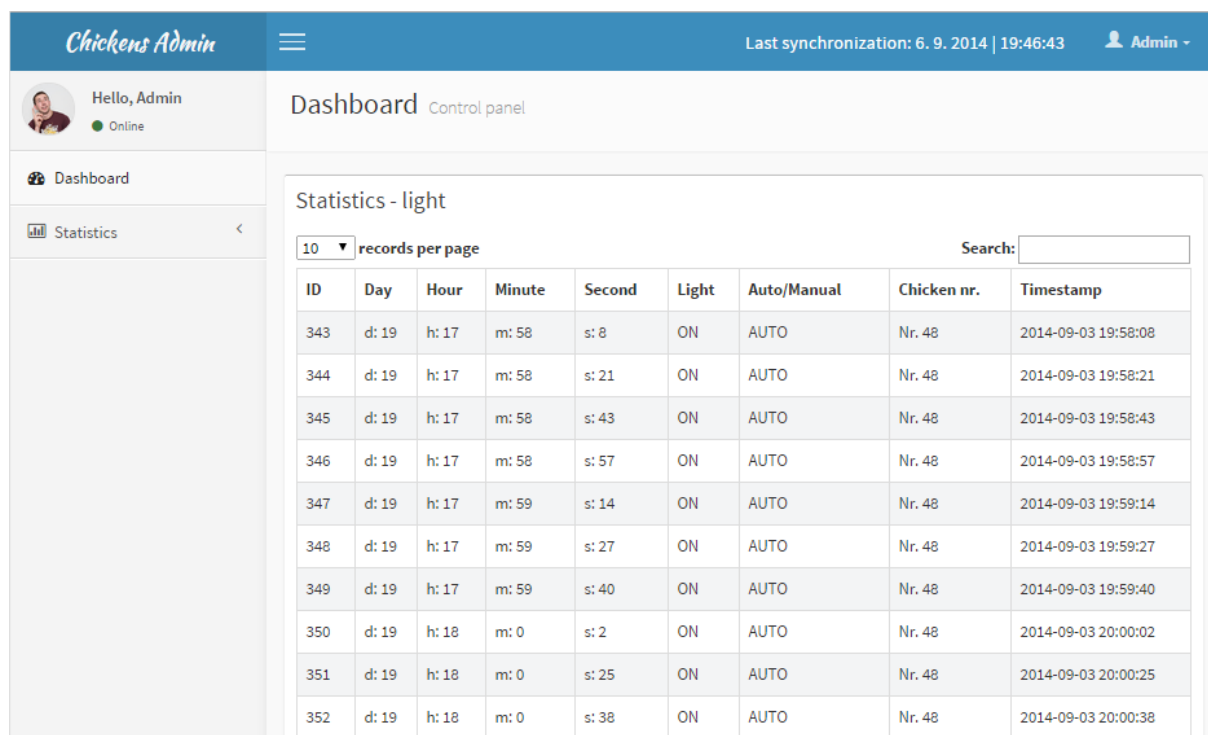
Znotraj skripte, ki predstavlja domačo stran naše nadzorne plošče, smo s pomočjo tehnologije PHP zgradili strukturo elementov HTML. Glede na trenutna stanja smo nastavili ustrezna poimenovanja razredov in identifikatorjev ter določili vsebino HTML elementov.

Za prikaz trenutnega stanja sistema je ponekod zadoščal zgolj prikaz trenutnih stanj, drugim pa smo dodali še gumbje za sprožanje različnih akcij.

Veliko pozornosti smo namenili tudi zagotavljanju odzivne oblike, zato smo znotraj dokumenta, ki definira stil CSS, poleg ostalih oblikovnih pravil, poskrbeli tudi za ustrezna oblikovna pravila naprav z nižjo resolucijo zaslona.

Za redno posodabljanje trenutnih stanj, smo napisali skripto v jeziku Javascript, ki je omogočila posodabljanje elementov HTML ter povzročila različne odzive na sprožene akcije uporabniškega vmesnika. Nastavili smo samodejno posodobitev elementov HTML na vsakih 5 sekund. Do trenutnih stanj pride skripta preko vrnjenega rezultata v obliki JSON (Slika 5.3). Akcije, ki povzročijo različne odzive, se med seboj razlikujejo, v večini primerov pa gre za klike z miško na različne HTML elemente.

Poleg prve strani nadzorne plošče smo ustvarili tudi več strani, ki so namenjene prikazu zabeleženih stanj naprav. Stanja smo prikazali znotraj tabel, ki omogočajo filtriranje med zabeleženimi zapisi (Slika 5.4).



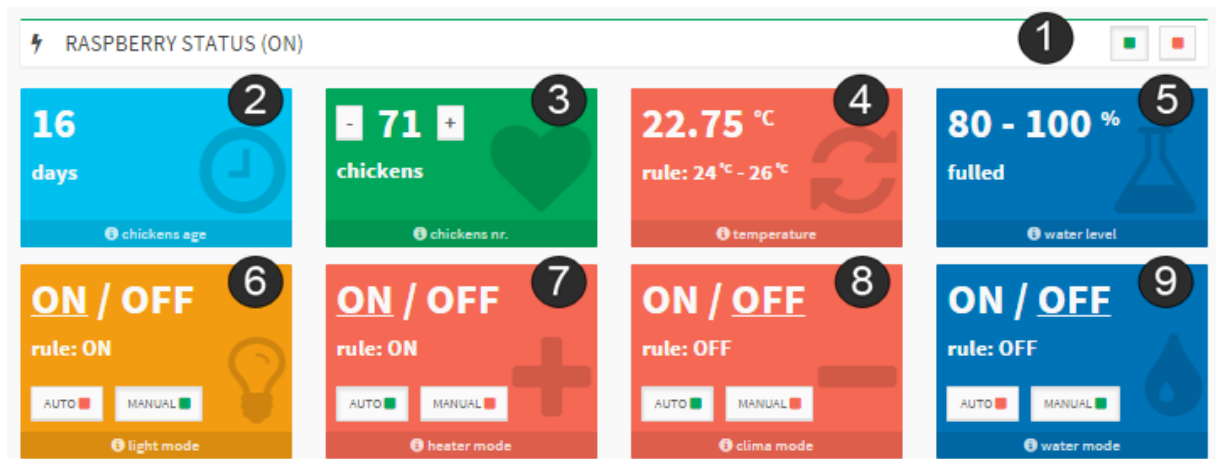
The screenshot shows the 'Chickens Admin' dashboard. The top header is blue with the title 'Chickens Admin', a menu icon, and the text 'Last synchronization: 6. 9. 2014 | 19:46:43' and 'Admin'. The left sidebar shows a user profile 'Hello, Admin' with 'Online' status and navigation links for 'Dashboard' and 'Statistics'. The main content area is titled 'Dashboard Control panel' and features a 'Statistics - light' section. This section includes a dropdown for 'records per page' set to '10' and a search bar. Below this is a table with 9 columns: ID, Day, Hour, Minute, Second, Light, Auto/Manual, Chicken nr., and Timestamp. The table contains 10 rows of data, all showing 'ON' status for the light and 'AUTO' for the mode.

ID	Day	Hour	Minute	Second	Light	Auto/Manual	Chicken nr.	Timestamp
343	d: 19	h: 17	m: 58	s: 8	ON	AUTO	Nr. 48	2014-09-03 19:58:08
344	d: 19	h: 17	m: 58	s: 21	ON	AUTO	Nr. 48	2014-09-03 19:58:21
345	d: 19	h: 17	m: 58	s: 43	ON	AUTO	Nr. 48	2014-09-03 19:58:43
346	d: 19	h: 17	m: 58	s: 57	ON	AUTO	Nr. 48	2014-09-03 19:58:57
347	d: 19	h: 17	m: 59	s: 14	ON	AUTO	Nr. 48	2014-09-03 19:59:14
348	d: 19	h: 17	m: 59	s: 27	ON	AUTO	Nr. 48	2014-09-03 19:59:27
349	d: 19	h: 17	m: 59	s: 40	ON	AUTO	Nr. 48	2014-09-03 19:59:40
350	d: 19	h: 18	m: 0	s: 2	ON	AUTO	Nr. 48	2014-09-03 20:00:02
351	d: 19	h: 18	m: 0	s: 25	ON	AUTO	Nr. 48	2014-09-03 20:00:25
352	d: 19	h: 18	m: 0	s: 38	ON	AUTO	Nr. 48	2014-09-03 20:00:38

Slika 5.4: Prikaz tabele zabeleženih stanj naprave za upravljanje svetlobe.

5.2.3.4 Predstavitev funkcionalnosti nadzorne plošče

Znotraj nadzorne plošče je možno nadzorovati trenutna stanja naprav. Nekatera stanja je mogoče tudi spreminjati, ostala pa služijo zgolj prikazu trenutnih stanj. Za lepši pregled nad stanji, smo posamezne dele nadzorne plošče obarvali različno. Zgolj zaradi preglednosti in zmožnosti opisa funkcionalnosti (Tabela 5.8), smo posamezne dele nadzorne plošče označili še s številkami (Slika 5.5). Prikazana stanja se osvežujejo na 5 sekund.



Slika 5.5: Prikaz vmesnika za ogled trenutnih stanj naprav ter zmožnost krmiljenja naprav.

Oznaka	Opis funkcionalnosti
1	Prikaz statusa našega sistema. V kolikor je v podatkovni bazi nastavljena vrednost 1 za polje <i>status</i> pomeni, da naj se naprave krmilijo preko našega zgrajenega sistema. V nadzorni plošči tako nastavljeno stanje pokažemo z izbranim gumbom zelene barve ter zgornjim zeleno obarvanim robom. V primeru, da imamo nastavljeno vrednost polja <i>status</i> na 0, pa to pomeni, da naj se naprave ne krmilijo preko našega sistema. V nadzorni plošči to prikažemo z izbranim gumbom rdeče barve ter z zgornjim robom rdeče barve. S klikom na gumb zelene barve nastavimo vrednost polja <i>status</i> na 1, s klikom na rdečo barvo pa na vrednost 0.
2	Prikaz trenutne starosti piščancev na podlagi nastavljenega cikla piščancev.
3	Prikaz trenutnega števila piščancev. S klikom na levi gumb z oznako minus od trenutnega števila odštejemo vrednost 1, s klikom na gumb z oznako plus pa trenutnemu številu prištejemo vrednost 1. Ob kliku na posamezen gumb, se osveži tudi prikazano trenutno število piščancev.

4	Prikaz trenutne temperature in pravila najnižje ter najvišje priporočene temperature, na podlagi pravil za trenutno starost piščancev.
5	Prikaz trenutnega stanja nivoja vode v obliki odstotkov napolnjenega zbiralnika.
6	Prikaz trenutnega stanja in pravila osvetlitve ter možnost nastavljanja samodejnega oziroma ročnega upravljanja luči.
7	Prikaz trenutnega stanja in pravila ter možnost nastavljanja samodejnega oziroma ročnega upravljanja električnega grelca.
8	Prikaz trenutnega stanja in pravila ter možnost nastavljanja samodejnega oziroma ročnega upravljanja električnega ventilatorja.
9	Prikaz trenutnega stanja in pravila ter možnost nastavljanja samodejnega oziroma ročnega upravljanja točenja vode.

Tabela 5.8: Razlaga funkcionalnosti nadzorne plošče sistema.

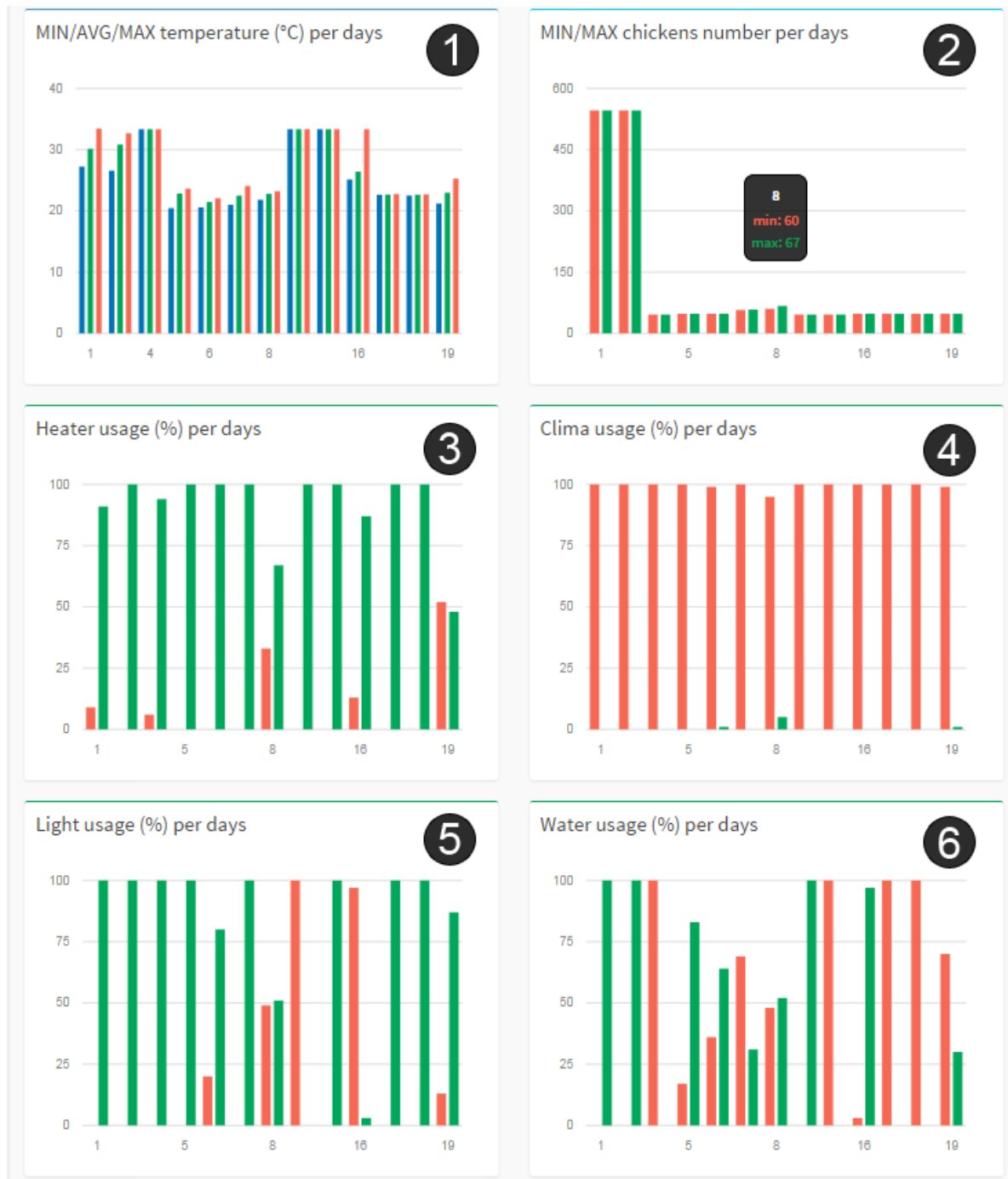
Funkcionalnostim, označenim s številko 6, 7, 8 in 9 (Tabela 5.8), je skupno to, da je podčrtana beseda *ON* v primeru, ko je v podatkovni bazi shranjeno stanje posamezne naprave na 1. Beseda *OFF* pa je podčrtana v primeru trenutno nastavljenega stanja z vrednostjo 0. Med stanji je mogoče preklapljati le takrat, ko je nastavljeno ročno spreminjanje stanj (aktiven je gumb z oznako *MANUAL*). Aktivnost gumba je ponazorjena z zelenim kvadratom znotraj gumba. Rdeč kvadrat znotraj posameznega gumba določa neaktivno stanje.

Poleg delov, ki prikazujejo trenutna stanja, smo na prvi strani nadzorne plošče izpostavili še šest grafov, izdelanih na podlagi zabeleženih stanj, shranjenih v tabeli *raspberry_statistics* (Slika 5.6). Grafi se med seboj razlikujejo po pomenu (Tabela 5.9). Predstavitev rezultatov in legenda, ki je prikazana znotraj grafa z oznako 2 v črnem pravokotniku (Slika 5.6), se prikaže, ko se z miško postavimo na enega izmed zabeleženih rezultatov.

Oznaka	Opis funkcionalnosti
1	Prikaz najnižje, povprečne in najvišje zabeležene temperature za posamezen dan starosti piščancev. Modra barva predstavlja najnižjo zabeleženo temperaturo, zelena povprečno, rdeča pa najvišjo.

2	Prikaz največjega ter najmanjšega števila piščancev zabeleženega za posamezen dan starosti piščancev. Rdeča barva predstavlja najnižje, zelena pa najvišje zabeleženo število.
3	Prikaz zabeležene uporabe električnega grelca v odstotkih za posamezen dan starosti piščancev.
4	Prikaz zabeležene uporabe električnega ventilatorja v odstotkih za posamezen dan starosti piščancev.
5	Prikaz zabeležene uporabe luči v odstotkih za posamezen dan starosti piščancev.
6	Prikaz zabeleženega stanja točenja vode v odstotkih za posamezen dan starosti piščancev

Tabela 5.9: Razlaga pomena prikazanih grafov nadzorne plošče sistema.



Slika 5.6: Prikaz grafov zabeleženih stanj naprav.

Poglavje 6 Sklepne ugotovitve

Ker pred izdelavo diplomske naloge računalnika RPI nisem poznal, me zelo veseli dejstvo, da nam je uspelo izdelati tako celovit sistem, ki je presegal vsa pričakovanja skrbnika. Poleg vseh funkcionalnosti, ki jih omogoča, smo presenetili tudi s cenovno ugodno rešitvijo, saj cena sistema ne presega 150 evrov. Pokazali smo, da je mogoče marsikatero opravilo optimizirati, kar nam daje elana za nadaljnje razširitve sistema. Glede na to, da sistem združuje tri dokaj različne nivoje, in sicer računalnik RPI, elektronsko vezje naprav in spletni vmesnik, so bile tudi naše težave, s katerimi smo se srečali, zelo različne.

Največ časa smo porabili za spoznavanje računalnika in vzpostavitev okolja za krmiljenje naprav. K sreči gre za vedno bolj priljubljen računalnik, ki temelji na odprtokodnem sistemu, zato se število knjižnic in orodij, ki nam lajšajo delo, stalno večja. Le te smo s pridom uporabili tudi pri naši nalogi.

Največjo odgovornost smo prevzeli z omogočanjem samodejnega krmiljenja točenja vode. Nismo si smeli privoščiti, da bi ta sistem kadarkoli odpovedal. S tem je imel skrbnik največ težav tudi v času ročnega krmiljenja sistema, zato smo temu namenili največ pozornosti. Krmiljenje luči in temperature nam ni povzročalo večjih problemov.

Naslednji večji izziv je bil vzpostavitev krmiljenja preko spletnega vmesnika. Ker smo veliko časa namenili načrtovanju sistema, in vse funkcionalnosti dodobra preučili pred implementacijo rešitve, večjih težav nismo imeli. Na podlagi predvidenih funkcionalnosti smo zgradili strukturo podatkovne baze, nato pa smo napisali še vmesnik API za delo s podatki. Pri izdelavi spletne nadzorne plošče smo imeli največ dela s prikazom trenutnih stanj in sprožanju akcij ob različnih dogodkih.

Računalnik RPI nas je povsem navdušil, predvsem zaradi ugodne cene, možnosti razširitve in zmogljivosti sistema, ki se stalno izboljšuje. Na trgu je na voljo ogromno dodatkov, ki jih pri izdelavi seminarske naloge nismo uporabili, dajejo pa sistemu še dodatne zmogljivosti. Poleg tega obstaja tudi veliko že zgrajenih projektov za povezavo in krmiljenje naprav preko spleta, in sicer velja omeniti vsaj WebIOPi [40], openHAB [41] in DeviceHive [42]. Vse omenjene projekte smo pri izdelavi našega sistema preizkusili, kasneje pa smo se, zaradi posebnosti našega projekta ter predznanja, odločili za lastno implementacijo rešitve.

Skozi izdelavo sistema smo spoznali tudi to, da smo si pustili še nekaj prostora za izboljšave. Ena bolj pomembnih je zagotovo ta, da bi uporabili računalnik RPI tudi kot strežnik, na katerem bi gostili našo spletno aplikacijo. V prostor bi lahko umestili še kamero za boljši pregled nad trenutnim stanjem. Še več pozornosti in dodatnega preverjanja trenutnega stanja bi namenili sistemu za točenje vode. Poskusili bi ugotoviti še način za krmiljenje hranjenja piščancev, katerega niti nismo nameravali avtomatizirati, zaradi dobro izpopolnjene trenutne rešitve, ki ne potrebuje stalnega nadzora. Na računalnik bi lahko povezali monitor za prikaz trenutnih stanj ter uredili možnost ročnega upravljanja s sistemom. Za še boljši pregled, pa bi izdelali bolj podrobna dnevna, tedenska in mesečna poročila ter poskrbeli za sms obveščanja ob kritičnih dogodkih.

Literatura

- [1] Pregled okvirnih normativov za pitanje piščancev Ross 308, Jata Emona (osebna korespondenca z družbo JATA EMONA d.o.o. Poslovna skrivnost).
- [2] Broiler Management Handbook Aviagen 2014. [Online:]. Dosegljivo na: http://en.aviagen.com/assets/Tech_Center/Ross_Broiler/Ross-Broiler-Handbook-2014-EN.pdf.
- [3] Raspberry Pi the complete manual, Aaron Asadi, Imagine Publishing Ltd, 2014.
- [4] Dr. Andrew Robinson & Mike Cook, *Raspberry Pi Projects*, John Wiley & Sons Ltd., 2014.
- [5] Upcoming board revision | Raspberry Pi. [Online:]. Dosegljivo: <http://www.raspberrypi.org/upcoming-board-revision/>.
- [6] Raspberry Pi Model B Revision 2. [Online:]. Dosegljivo: <http://southgoldcoastscan.net/img/raspberry-pi-model-b-rev-2-512mb.jpeg>.
- [7] High Definition 1080p Embedded Multimedia Applications Processor – BCM2835 | Broadcom. [Online:]. Dosegljivo: <http://www.broadcom.com/products/BCM2835>.
- [8] Raspberry Pi Architecture. [Online:]. Dosegljivo: <http://meseec.ce.rit.edu/551-projects/fall2013/2-1.pdf>.
- [9] Eben Upton & Gareth Halfacree, *Raspberry Pi User Guide*, John Wiley & Sons Ltd., 2012.
- [10] Downloads | Raspberry Pi. [Online:]. Dosegljivo: <http://www.raspberrypi.org/downloads>.
- [11] Image Writer project files : Image Writer. [Online:]. Dosegljivo: <https://launchpad.net/win32-image-writer/+download>.

- [12] Matt Richardson in Shawn Wallace, Getting Started with Raspberry Pi, O'Reilly Media, Inc, 2012.
- [13] PuTTY Download Page. [Online] Dosegljivo:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.
- [14] Controlling GPIO with wiringPi. [Online] Dosegljivo:
<http://raspberrypiwebserver.com/gpio/wiringPi.html>.
- [15] 15.1. os – Miscellaneous operating system interfaces Python 2.7.8 documentatiton. [Online:]. Dosegljivo na: <https://docs.python.org/2/library/os.html>.
- [16] 10.7. glob – Unix style pathname pattern expansion Python 2.7.8 documentation [Online:]. Dosegljivo na: <https://docs.python.org/2/library/glob.html>.
- [17] RPi GPIO 0.5.7 : Python Package Index [Online:]. Dosegljivo na: <https://pypi.python.org/pypi/RPi.GPIO>.
- [18] 19.2. json – JSON encoder and decoder Python 3.4.2 documentation [Online:]. Dosegljivo na: <https://docs.python.org/3/library/json.html>.
- [19] 20.6. urllib2 – extensible library for opening URLs Python 2.7.8 documentation [Online:]. Dosegljivo na: <https://docs.python.org/2/library/urllib2.html>.
- [20] 13.1. csv – CSV File Reading and Writing Python 2.7.8 documentation [Online:]. Dosegljivo na: <https://docs.python.org/2/library/csv.html>.
- [21] 15.3. time – Time access and conversions Python 2.7.8 documentation [Online:]. Dosegljivo na: <https://docs.python.org/2/library/time.html>.
- [22] 8.1. datetime – Basic date and time types Python 2.7.8 documentation [Online:]. Dosegljivo na: <https://docs.python.org/2/library/datetime.html>.
- [23] Temperature Measurements. [Online] Dosegljivo:
<http://www.sbprojects.com/projects/raspberrypi/temperature.php>.
- [24] Arduino-info – ArduinoPower [Online:]. Dosegljivo na: <http://arduino-info.wikispaces.com/ArduinoPower>.

- [25] Electronics DIY (For Arduino) 5V Relay Module [Online:]. Dosegljivo na: http://www.miniinthebox.com/electronics-diy-arduino-5v-relay-module_p340896.html.
- [26] Keyes 5V Relay Module. [Online] Dosegljivo: <http://www.dx.com/p/arduino-5v-relay-module-blue-black-121354>.
- [27] Brendan Horan, *Practical Raspberry Pi*, Technology in action, 2013.
- [28] One Wire Digital Temperature Sensor – DS18B20 [Online:]. Dosegljivo na: <https://cdn.sparkfun.com//assets/parts/1/8/7/00245-2.jpg>.
- [29] Liquid Water Level Sensor Horizontal Floating Float Switch [Online:]. Dosegljivo na: http://m4.uxcell.com/photo_new/20101130/c/ux_a10113000ux0241_ux_c.jpg.
- [30] 50 Series Vertical Float Switch. [Online]. Dosegljivo: <http://www.deeterelectronicsinc.com/products/float-switch-liquid-level-sensors-overview/50-series-vertical-liquid-level-sensors-float-switch>.
- [31] 12V DC Electric Solenoid Valve Water Air N/C 1/2" Normally Closed Brass Filtered TFW-1S Imported. [Online]. Dosegljivo na: <http://www.tanotis-mart.com/12v-dc-electric-solenoid-valve-water-air-nc-12quot-normally-closed-brass-filtered-tfw-1s-imported>.
- [32] Solenoid water valve TWS-1 [Online:]. Dosegljivo na: [http://i.ebayimg.com/00/s/MTAwMVgxMDAx/z/-RoAAMXQtUxTdbGO/\\$_57.JPG](http://i.ebayimg.com/00/s/MTAwMVgxMDAx/z/-RoAAMXQtUxTdbGO/$_57.JPG).
- [33] Robin Nixon, *Learning PHP, MySQL, JavaScript, CSS & HTML5 - Third Edition*, O'Reilly Media, 2014.
- [34] jQuery. [Online]. Dosegljivo: <http://jquery.com/>.
- [35] JSON. [Online]. Dosegljivo: <http://json.org/>.
- [36] AJAX Introduction. [Online]. Dosegljivo: http://www.w3schools.com/ajax/ajax_intro.asp.
- [37] Bootstrap. [Online]. Dosegljivo: <http://getbootstrap.com/>.
- [38] Installation – Slim Framework. [Online]. Dosegljivo: <http://www.slimframework.com/install>.

- [39] Free Admin Template For Bootstrap – AdminLTE | Almsaeed Studio. [Online]
Dosegljivo: <http://www.almsaeedstudio.com/>.
- [40] Webiopi – Internet of Things framework – Google Project Hosting. [Online].
Dosegljivo: <https://code.google.com/p/webiopi/>.
- [41] DeviceHive – M2M, Machine-to-Machine Communication Framework. [Online].
Dosegljivo: <http://www.devicehive.com/>.
- [42] openHAB – empowering the smart home. [Online:]. Dosegljivo:
<http://www.openhab.org/>.